# Predictive Monitoring of Business Processes: A Survey

Alfonso Eduardo Márquez-Chamorro, Manuel Resinas, and Antonio Ruiz-Cortés

**Abstract**—Nowadays, process mining is becoming a growing area of interest in business process management (BPM). Process mining consists in the extraction of information from the event logs of a business process. From this information, we can discover process models, monitor and improve our processes. One of the applications of process mining, is the predictive monitoring of business process. The aim of these techniques is the prediction of quantifiable metrics of a running process instance with the generation of predictive models. The most representative approaches for the runtime prediction of business process are summarized in this paper. The different types of computational predictive methods, such as statistical techniques or machine learning approaches, and certain aspects as the type of predicted values and quality evaluation metrics, have been considered for the categorization of these methods. This paper also includes a summary of the basic concepts, as well as a global overview of the process predictive monitoring area, that can be used to support future efforts of researchers and practitioners in this research field.

**Index Terms**—Business process management, process mining, predictive monitoring, process indicators

## 1 Introduction

Process mining techniques allow the extraction of useful information from the event logs and historical data of business processes (BPs) [62]. This information can help to improve the processes and is generally extracted after the process has been finished. However, the interest to apply process mining to running process instances is increasing.

Predictive monitoring of BPs [59] is one of the sub-fields of process mining and aims to provide timely information that enable proactive and corrective actions to improve process performance and mitigate risks. It can be defined as the set of runtime methods aimed at generating predictive models [26] that can be used for the prediction of a particular value of a process instance given its ongoing trace and the event log of historical traces as inputs. As input of these methods, the event log provides the necessary characteristics which define the process for the prediction. Additionally, a complete process model, such as a Petri net (PN), or contextual attributes have been optionally considered as input data. As output of the methods, a predicted value for each running process instance or collection of them is obtained. This value belongs to a given domain, and may be boolean, categorical or numerical depending on the object of prediction, e.g., the remaining time of a process (numeric) or the fulfillment of a certain goal (boolean). Thus, the development of mechanisms to predict these values based on the runtime processing of the event streams exchanged between different information systems is very appealing from a practical standpoint. These predicted values can be metrics or process indicators evaluating the performance of a BP in terms of efficiency and effectiveness, or help to evaluate risks or predict possible service level agreement (SLA) violations.

In the last six years, a variety of different approaches for predictive monitoring have appeared. They have been developed to predict different kinds of metrics, have faced the problem from different angles and have been applied to different domains. However, despite their differences, they all share many commonalities. Therefore, a joint analysis of all these approaches can provide an overall view of them as well as identify new challenges in this field. This is the main goal of this survey, which collects and analyzes a compilation of runtime monitoring prediction approaches on BPs. These relevant and ultimate methods include techniques based on machine learning approaches, statistical methods, annotated transition systems and hybrid methods. Furthermore, from this analysis, we identify the most relevant concepts that compose a predictive monitoring approach and discuss the ways the different approaches are tackling each of them.

Two other issues closely related to the predictive monitoring has received a lot of attention in recent years. First, process deviance mining aims at explaining the deviance cases of a process instance [41]. Deviance mining techniques use both normal and deviant traces as input, and returns a set of rules to give the reasons of the possible deviations. Two are the main differences with respect to predictive monitoring, namely: 1) whereas the prediction is performed for an ongoing process instance in real time, the deviance analysis is performed after the execution of processes;

- *A.E. Márquez-Chamorro is with the Departamento de Lenguajes y Sistemas Informáticos, University of Seville, Sevilla 41004, Spain. E-mail: amarquez6@us.es.*
- *M. Resinas is with the Universidad de Sevilla, Sevilla 41004, Spain. E-mail: resinas@us.es.*
- *A. Ruiz-Cortés is with the Computer and Languages Systems, University of Seville, Sevilla 41004, Spain. E-mail: aruiz@us.es.*

2) whereas predictive monitoring uses incomplete ongoing traces for the prediction, deviance mining analyses the complete traces of normal and deviant cases. In the latter case, detection of failures of software systems has been previously considered in [51]. Both approaches, predictive monitoring and detection of failures, have similarities since the input data and the output can be similar for some cases. However, they have two main differences. First, despite the similarities in the input data, they are applied to different domains (software systems and BPs) with different characteristics. Second, each approach has aspects not covered by the other. Specifically, predictive monitoring also considers other possible objects of prediction such as time predictions, probability of a risk or prediction of the next event, among others that are not relevant for the detection of failures of software systems, whereas the detection of failures of software systems include aspects such as failure tracking and undetected error auditing that are not applicable as is in predictive monitoring of BPs.

Our study can be used to support future efforts of practitioners and researchers in the predictive monitoring field. On the one hand, practitioners can use the concepts identified as a framework on which a predictive monitoring system for BPs can be built. Furthermore, the discussion on the approaches described can help to identify the techniques from which they can choose to implement the one that better suites their needs. On the other hand, researchers will obtain twofold support from this analysis. First, the concepts identified and the overall view provided may help guide research efforts on new predictive models that improve the performance of current approaches. Second, new researchers in this area will get a global overview on what is done currently in the field and which are the open challenges that require more research.

The rest of the paper is organized as follows. Section 2 summarizes some basic concepts in the area of predictive process monitoring. In Section 3, the most relevant techniques are described. Section 4 discusses how the current techniques deal with the different steps involved in predictive process monitoring. Finally, Section 5 concludes the paper and identify open challenges in this field.

## 2 Preliminary Concepts

In this section, several useful concepts in the area, as well as the review method considered for the survey, are explained. The general methodology for the predictive monitoring of BPs used in the majority of the papers is presented in Section 2.2. An introduction of input data for the prediction, databases, encoding, checkpoints and the experimental validation are shown in Sections 2.3, 2.4, 2.5, and 2.6 respectively. Finally, Sections 2.7 and 2.8 introduces the different objects of prediction according to different dimensions (Section 2.7) and, specifically, according to their application domain (Section 2.8).

### 2.1 Review Method

Existing literature in predictive monitoring of BPs was searched in the online repositories of the main technical publishers, including Scopus, Web of Science (WOS) and Google Scholar. As inclusion criteria, we have incorporated those papers since 2010 that addresses any topic related predictive monitoring of BPs, have been cited at least 5 times (this number of cites have been considered in other surveys to indicate relevant papers), and were published in indexed journals, relevant conferences[1] and other books and conferences in the area. The 5-cited constraint was omitted for those works published between 2015 and 2017, assuming that, due to a shorter period of time, they have not yet achieved this number of cites. As exclusion criteria, we have excluded those papers not related to the computer science field, not written in English, or not accessible on the Web.

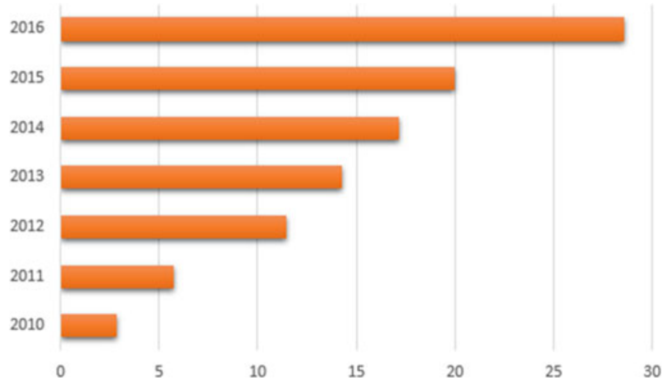

Fig. 1. Percentage of published papers per year.

Specifically, we collected computer science papers since 2010 that have either "predictive monitoring" AND "business process" (search string 1) or "business process" AND "prediction" (search string 2) in their keywords, title or abstract. We have chosen these search strings because these keywords appear consistently in the most relevant related work on predictive monitoring. SCOPUS provided 10 results using the search string 1: TITLE-ABS-KEY ("business process" AND "predictive monitoring"), and 195 results using using the search string 2: TITLE-ABS-KEY ("business process" AND "prediction"). Filtering by number of cites, we have obtained 37 works from SCOPUS. Additionally, using the same search settings, WOS returns 4 results for search string 1 and 166 results for the search string 2. Only 35 works has more than or equal to 5 authors.[2] Finally, the same searches were reproduced in Google Scholar, obtaining 199 results for the search strings and considering only the ten first pages for our work. After filtering by the required topic and removing the repeated papers in the different searches, a total of 41 publications were finally considered on the scope of our review. Next, we examined the abstracts of the papers identified in the previous step and filtered them according to the predicted values and types of the different methods.

Fig. 1 shows the percentage of published paper from 2010 to 2016. A general upward trend in the number of publications in this area is observed. Two works published in 2017 were also collected in this survey. The distribution of the papers according the publication venue was also considered. A 51,85 percent of the presented papers belong to indexed journals. The relevant conferences represent the 37,03 percent of the total of papers. A percentage of 11,11 percent corresponds to books and other conferences.

1. The CORE ranking of conferences has been considered (http://www.core.edu.au/conference-portal).
2. Results of SCOPUS and WOS searches are collected in: goo.gl/r3Qdu7.

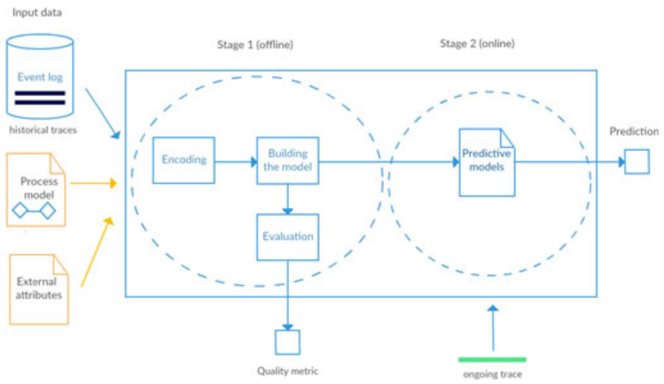


Fig. 2. Experimental procedure of a general predictive monitoring method.

## 2.2 Predictive Monitoring Methodology

This section describes a general methodology for the predictive monitoring of BPs. Several differences exist between predictive monitoring and other types of prediction tasks, we point out the following. First of all, process-aware methods are clearly distinct because they use techniques based on the process structure, such as annotated transition and graph-based systems. Second, predictive monitoring is carried out at real-time during the execution of the process instances in a certain period. This implies that the prediction is made at a certain point of the execution, which is named checkpoint. This also affects other factors, such as the evaluation and the creation of the predictive models. Different selection strategies to define the checkpoints have been considered [24], [40]. Finally, even if general machine learning techniques are used such as decision trees (DTs), an adequate encoding of the event log must also be considered. Encoding usually involves a feature engineering task which is always specific to the concrete process and hinder the initial stages of the predictive monitoring process.

Stage 1 of Fig. 2 represents the learning phase. In this stage, the event log of a process and, optionally a process model and additional external information will constitute the input data of the predictive monitoring method. These input data are then generally encoded in feature vectors that can be interpreted by the predictive algorithm. Then, the predictive method is executed and generates a prediction model as output data, based on the knowledge of the traces of the event log. This model is evaluated to asses its validity, using the different traces of process instances as a test set, by means of quality metrics.

Stage 2 of Fig. 2 represents the prediction phase for a typical predictive monitoring method. At run-time, the generated model is applied to ongoing instances in a given moment of the execution. Then, the predictive model will determine the value of the predicted outputs for this process instance. It should also be noted that the majority of predictive monitoring techniques collected here consists of an offline and an online component that corresponds to Stage 1 and Stage 2, respectively. In many cases, the offline component, which deals with the generation of the predictive model, is computationally expensive, but the online component (Stage 2), which addresses the predictions based on the the generated model is fast. Fig. 2 presented the general stages of a predictive monitoring process that are common to most approaches. However, for specific approaches, each step can be decomposed in more detail. For instance, [36] provides a detailed methodology of predictive monitoring processes based on machine learning approaches.

TABLE 1
Event Log Example

| case id | event id | timestamp | ev. | resource | cost |
|---|---|---|---|---|---|
| 1 | 107561 | 12-12-2016:12.15 | A | Lucas | 100 |
| | 107562 | 12-12-2016:14.55 | B | Lucas | 300 |
| | 107563 | 12-12-2016:17.30 | C | Paul | 200 |
| | 107564 | 13-12-2016:12.15 | D | Laura | 400 |
| 2 | 108631 | 14-12-2016:10.00 | A | Fred | 100 |
| | 108632 | 14-12-2016:12.52 | D | Fred | 200 |
| | 108633 | 14-12-2016:13.27 | E | Barney | 100 |
| 3 | 108945 | 15-12-2016:10.32 | B | Alan | 100 |
| | 108946 | 16-12-2016:09.18 | E | Sylvia | 300 |

## 2.3 Input Data

The main input data for the predictive monitoring methods is the event log. Table 1 represents a general log of a process where each row represents the execution of an activity of the process and its information. Typically, this information consists of the identifier of the process instance and event and the timestamp where the event was executed. Additional information can also be included in the log, such as the name of the resource who execute the activity or the cost of the activity.

These event logs are generally provided by information systems that record traces about process executions. Massive amounts of information can be generated by one of these systems which are stored in event logs. As a consequence, it is necessary to acquire the more relevant process characteristics for the data management following the classification described in [16], according to four different perspectives. First, the control-flow perspective, related to the order of the activities performed in the process. Second, the data-flow perspective which involves the different attributes attached to the events. Third, the time perspective which is related to various types of duration in the process, such as the duration of an activity or the remaining time of a process. Finally, the resource/organization perspective related to the resource that executes a determined event [7]. These perspectives can be appreciated in the different columns of the log example (Table 1): an event id, which is a unique identifier of each event, a timestamp (time perspective), that indicates the time and date of the execution of an activity, the name of this activity (control-flow perspective), the resource or person who executes the activity (organization perspective), the cost of the activity and other useful information about the event (data-flow perspective). Some of the gathered works [49], [63] also include as input data, a complete process model, represented by for example, a Petri net. However, this does not mean that the process model has to be provided by the user, the model can also be discovered automatically using process mining techniques like in [31]. External or contextual attributes have been also considered as input data (e.g., the weather).

## 2.4 Encoding

Before building the model, it is necessary to describe an encoding which stores enough information of the process, that will be used as input for the technique employed to build the model. Generally, the encoding for a trace includes only the flow perspective. The data-flow perspective is also incorporated in some encodings, considering the information data of the events and not only the sequence flow. The encoding usually represents the events and their associated information. Different sizes of the historic of events can be considered in the encoding, e.g., some approaches take into account only the last event, a few number of events or the complete process. In addition, in some cases, metrics such as the number of resources involved in a process instance, are computed from the events to provide additional information for building the model. Finally, this step also includes the computation of the value of the metric to be predicted. This value is computed according to the existing attributes of the historical traces, e.g., as a result of a combination or arithmetic operation between two or more properties or as the evaluation of a LTL formula.

### 2.5 Building the Model

Several predictive models can be considered according to the type of object to be predicted. Three examples are cited in the following: A decision or regression tree can be useful to determine a discrete or continuous value of a particular output. Decision or association rules can show different situations for the risk predictions. An annotated transition system can be valid for the prediction of the time completion of a process.

The methods used for building the model can be classified according to its process awareness. A predictive model is process-aware if it exploits an explicit representation of the process model to make the prediction (e.g., an annotated transition system, or a stochastic Petri net). Instead, a non-process-aware predictive model do not use an explicit representation of the process model (e.g., DTs). The process model used in process-aware methods are either provided as input or obtained using a process discovery technique from the event log.

Furthermore, some models need the indication of checkpoints [33], where the prediction is carried out. These points are necessary for machine learning approaches but not for the annotated transition system, because they gather the information about the complete process. Each one of the checkpoints should be established before an activity in a BP. For each checkpoint, a predictive model has to be generated for the predictive method. Some of the strategies for the selection could be the choice of checkpoints after each executed decision activity or to establish a checkpoint for each activity that exceed the mean execution time. Selection strategies to define the checkpoints is considered in [69].

### 2.6 Evaluation of the Model

For the accuracy assessment of predictive methods, works in the area have considered the type of method for the prediction (classification or regression) depending on the object of prediction. In the case of classification methods, for the prediction of boolean or categorical values, it is natural to use classification measures: Precision represents the number of correctly predicted process instances, while recall reflects the proportion of predicted process instances divided by the total number of instances. Thus, $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+TN}$ where $TP$ is the number of correct predictions (true positives), $FP$ is the number of predicted false positives and $FN$ represents the number of false negatives. Therefore, $TP + FN$ represents the total number of process instances and $TP + FP$ reflects the total of predicted instances. Some works also determine the accuracy of the methods where $accuracy = \frac{TP+TN}{TP+FP+TN+FN}$. Accuracy represents the proportion of correctly classified results (both positive and negative). Furthermore, other reliable indicator is Area Under Curve (AUC). AUC provides a single measure of a classifier performance and allows the visualization of the trade-off between the true positives rate (recall) and the false positive rate. False positive rate (FPR) is equal to FP/ FP+TN and represents the cost of the algorithm. In a AUC diagram, the diagonal line represents a random classifier. Points above the diagonal represent good classification results (better than random). Points below the diagonal represent poor classification results (worse than random). AUC is more resilient to class imbalance and takes into account the likelihood scores. The AUC provides a single measure of a classifier performance for evaluating which model is better on average. It allows the visualization of the trade-off between the true positives rate and the false positive rate

In the case of regression methods, for the prediction of numerical values, measures of quantitative deviance are employed, such as Root-mean Squared Error (RMSE), which calculates the error between the real and the predicted values. Finally, other authors utilised Mean Absolute Error (MAE) which implies more resilience to outliers. Thus, the formulas of RMSE and MAE are, respectively: $RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_i - y_i')^2}$ and $MAE = \frac{1}{n}\sum_{t=1}^{n}(y_i - y_i')$, where $y$ represents the real value, $y'$ represents the predicted value, and $n$ indicates the total number of instances. In this sense, other variations of the cited measures are applied in the literature, such as Root-mean square percentage error (RMSPE), Square root of the mean square error (sRMSE) or Mean absolute error.

### 2.7 Predictions

The predictions obtained are any kind of value that can be computed from the event log. Some examples are the next activity that is executed in the process instance, the fulfillment of linear temporal logic (LTL) constraints, the remaining time of the process instance, or a risk associated to the appearance of a specific value in a data object of the process instance. These predictions can be classified attending to three dimensions.

Attending to the prediction value, predictions can be classified into two broad categories depending on whether the object of prediction is a categorical or a numerical value. This classification is useful because the methods employed to build the model and the metrics to evaluate the model usually depend on these categories.

Attending to the scope of the prediction, the value predicted can refer only to one process instance, e.g., the remaining time of the process instance, or it can be an aggregation of several process instances, e.g., the average cycle time of all process instances that finished this month. Only two proposals in the collected works deal with aggregations, the other focus on predictions for one process instance.

Finally, attending to the domain to which the prediction is applied, the collected works cover four different domains: performance indicators, risk predictions, SLA violation predictions and other predicted values.

### 2.8 Application Domains

As we have stated in the previous section, we can classify the different predictions according to its application domain, i.e., performance indicators, risk predictions, SLA violation predictions, and other predicted values.

Performance requirements of a BP are specified through process performance indicators (PPIs). In general, PPIs are defined as "quantifiable metrics that allow us to evaluate the efficiency and effectiveness of a process" [17], [18]. They are aimed to control and improve the process. A PPI reflects the "critical success factors of a BP defined within an organisation, in which its target value reflects the objectives pursued by the organisation with that BP" [19]. We can consider the categories defined in [18]: time, count, data, state or derived indicators. Time is one of the most valuable indicators during the execution of a BP. Generally, time indicators measure the processing time from a start point to an end point of the process execution. The duration of an activity, the average lifetime or the time to completion of a process [42], [64] are other time indicators predicted in the literature. Since time is a continuous value, regression methods are employed for its prediction. Regarding the literature, process-aware systems are mostly used for the forecasting of time. In addition RMSE is the most commonly evaluation metric to asses the performance of the time prediction methods.

A risk prediction provides information about an specific risk and is used as a warning system for future actions. These statistics or measurements are revised periodically to alert the company about the changes that may indicate possible risks. Among the possible risks of a running instance of a process we can consider an abnormal execution time or multiple activity repetitions. In these cases, the training data is often very unbalanced, due to the fact that normal instances are much more usual than abnormal ones. AUC and F-score which provide a trade-off between the true positives rate and the false positive rate should be used for evaluations. Classification methods are used for the prediction of risks, since these predicted values are often discrete values.

Service level agreements (SLAs) define a contract of a determined service between a provider and a customer [11]. SLA violations must be avoided to prevent penalty payments and to enhance the customer satisfaction [33]. The predictions are then used to identify whether a SLA will be violated. Classification methods are generally employed for the SLA predictions.

Other predicted values, such as the abnormal termination or the prediction of the next event of a BP running instance, does not fit into any of the previous categories. However, they provide relevant information for the BP management and are also considered in the survey. Specifically, next event prediction appears commonly in predictive monitoring works. Statistical methods, as prediction techniques, and accuracy and precision, as evaluation measures, are generally applied in the literature for this type of prediction.

It is important to remark that the type of prediction value and the scope of the prediction have an influence on the method used and the predictive model built. However, this is not the case for the prediction domain. Many proposals in the literature are not tailored to a specific prediction domain, but they can be applied to many different domains if the value predicted (categorical or numerical) is the same. For instance, an approach that relies in DTs can predict any categorical value regardless of whether it is the next event, or the fulfilment of a performance indicator, or the chance a risk appears.

## 3 Methods

Existing techniques for predictive BP monitoring have been classified according to the process-awareness of the methods, i.e., whether the methodology exploits an explicit representation of the process model to make the prediction or not, and the type of problem, i.e., classification or regression, based on the type of predicted value (categorical or numerical).

### 3.1 Process-Aware Approaches

All the process-aware methods and their achieved results are summarized in Table 2. First columns indicates the author, year and the name of the proposal (if exists). Second column shows the reference of the work. Third and forth columns represent the quality assessment value and the quality measure. Fifth column shows a description of the dataset for the experimentation. Finally, sixth and seventh columns present the type of methodology for each proposal and the problem which try to solve (type of prediction), respectively. This table structure is also followed in the rest of sections.

#### *3.1.1 Regression Methods*

Among the process-aware regression methods, 3 proposals are based on machine learning and 8 are based on annotated transition systems (ATS) and statistical methods. In the majority of cases (7/11) they use RMSE as quality measure and best results are achieved by [10]. Real scenario datasets are employed in 9/11 cases, and public datasets are used in [39] and [49]. Finally, the software of 2 proposals are available ([63] and [55]).

First, we have considered ATS for the prediction of process indicators. A first approach, defined in [48], predicts the remaining process execution time, using the analysis of stochastic PNs with distributed transitions (GDT_SPN). The sequence of events and the time distributions are included in this model. The predictions are based on non-parametric stochastic models and parametric models obtained from historical event logs. As inputs, the method receives the GDT_SPN model of the BP, the ongoing trace of the process instance up to current time, the current time and the number of simulation iterations. The algorithm returns the average of simulated completion times of each iteration. This method is implemented as a plugin of ProM process mining tool. The second work, described in [49], presents a method based on non-Markovian PNs. These PNs are enriched with duration distributions (transition durations) and probability of firing (decision probabilities) for each transition. This information is obtained from the historical traces. The method predicts both the execution time and the risk of reaching a temporal deadline. This method uses the

TABLE 2
Summary of Process-Aware Methods for Regression and Classification Problems

| Author, Year (Name) | Ref | Q | Q Type | Dataset | Method | Prediction |
|---|---|---|---|---|---|---|
| Regression methods | | | | | | |
| Aalst, 2011 (FSM Analyzer) | [63] | 4.92 | RMSE | 796 cases | TS | Time |
| Metzger, 2012 (FInest) | [39] | - | - | 23,000 instances | Stat. | Delays |
| Rogge-Solti, 2013 (Stoch. PN) | [48] | 4.5 | RMSE | 784 cases | Stoch. models | Time |
| Polato, 2014 | [46] | 1.87 | RMSE | 5,000 traces | SVM, NB | Time |
| Rogge-Solti, 2015 | [49] | 2.5 | RMSE | BPI Ch. 2012 | GDT_SPN | Time |
| Senderovich, 2015 | [55] | 15 | RASE | 7,000 instances | Queueing theory, ATS | Delays |
| Polato, 2016 | [47] | 6.06 | RMSPE | Road fines log | LTS, SVM, NB | Time, next event |
| Cesario, 2016 | [10] | 0.2 | RMSE | 5,336 log traces | Clust, GSPN, Cloud | Time |
| Senderovich, 2016 (P3-FOLD) | [53] | 125 | sRMSE | US Hospital | ILP, GSPN | Time |
| Song, 2016 | [60] | - | - | Sintethic dataset | Graph-based system | Cost |
| Lakshmanan, 2015 | [31] | - | - | 2,000 traces | PPM, Markov chain | Prob. next event |
| Classification methods | | % | | | | |
| Becker, 2014 | [1] | 71.4 | Accuracy | 3,777 instances | PFA | Next event |
| Breuker, 2014 | [4] | 73.5 | Accuracy | BPI Ch. 2012 | EM | Next event |
| Metzger, 2014 | [40] | 68.3 | Precision | 3,942 traces | QoS agregg. rules, ANN | Risk |
| Breuker, 2016 (RegPFA) | [5] | 81.1 | Accuracy | BPI Ch. 2012, 2013 | PN, PFA | Next event |
| Conforti, 2016 | [14] | 86.72 | Accuracy | 9,350 instances | PING | Risk |
| Unuvar, 2016 | [61] | 91.0 | Accuracy | Marketing campaign | DT | Next event |

prediction time to improve the accuracy and evaluate the model against other similar approaches. The prediction algorithm takes five inputs: the GDT_SPN model, the current time, the deadline, the ongoing trace of the case and the number of iterations of the algorithm and returns the predicted remaining time of the process. A good state of the art in time predicting systems and predictions of risks is also described in this work. In the third work, an ATS is presented in [63] for the prediction of time completion of a process. This tool, named FSM Analyzer, receives a transition system and an event log as input data, and returns a transition system with extended information useful for the predictions. This information consists of average, standard deviation, minimum, and maximum remaining time for each state of the transition system. This method is integrated as a ProM plug-in.[3]

Machine learning approaches for regression are considered in the following works. First, authors present a machine learning approach in [46] a Naive-Bayes (NB) and SVR approach which predicts the remaining time of a running process. This estimation considers the probability of the future states of a transition system calculated by a NB technique and the estimated duration of the process obtained by the model. The output is the remaining time prediction. An extension of this work, presented in [47], considers stable and dynamic processes to predict remaining time and the future sequence of activities. A set of machine learning approaches, such as Naive Bayes and SVR, performs annotation over a labeled transition system (LTS) for the prediction. The input of the method consists of the process control flow and additional attributes of the events. Finally, in [10], a cloud-computing platform is described in this method. The implementation, which follows a trace clustering scheme and a regression method for the prediction of the remaining processing time, is based on a cloud based service-oriented infrastructure to allow the computation of enormous event logs. Several event attributes, including context features, are considered.

In [31], authors developed a statistical method for the prediction of probabilities of occurrence of the next event. They have obtained an instance-specific probabilistic process model (PPM) which can be converted to a Markov chain to determined the cited probabilities, returned as output of the method. The event log and a discovered process model are used as inputs.

The method showed in [39] applies statistical techniques for the prediction of events and their correlation with contextual elements of transportation processes, such as the weather conditions or road traffic. An integration platform named FInest, that incorporates the predictive monitoring module, was performed. The method receives three different sources of data: system messages from the processes, aggregates data with additional information of the processes, such as estimated time of arrival versus actual arrival or the cause for delays, and quality indicators. The system returns a prediction of the delay in the deliveries.

In [53], authors pose a simplification of the PNs models for the improvement of performance prediction. Their method, named P3-fold, generates several simplification rules for this task, using Integer Linear Programming (ILP). An initial generalized stochastic Petri net (GSPN) is received by the algorithm.

A cost analysis and prediction technique for manufacturing processes is presented in [60]. The control flow perspective, production volume and time are used as input of the algorithm. A graph-based system estimates the remaining cost of the ongoing process.

Finally, queueing theory and regression-based techniques are combined for the delay prediction in [55]. This work considers the queuing perspective of process, which represents the delayed execution time due to queueing effects. An annotated transition system is employed as input of the method. Some previous works of these authors also analyze queue mining techniques for the

3. processmining.org

predictive monitoring [52], [54]. The software of this proposal is available.[4]

### 3.1.2 Classification Methods

There are 2 process-aware classification methods that are based on machine learning, other 2 are based on annotated transition systems and graphs and other 2 are based on statistical methods. In the majority of cases (5/6) they use accuracy as quality measure and [61] obtained the best results (91 percent). Available datasets are employed in 3 works ([4], [40] and [5]). Finally, the software and data of two proposals can be downloaded ([1] and [40]).

Probabilistic models, such as stochastic models, are included in the two next works. The approach presented in [4] determines an analytic model to predict the following steps of a running process instance based on the probabilities returned by a expectation maximization (EM) method. This approach also incorporates a probabilistic finite automaton (PFA) as process discovery algorithm, to encode the log events. Finally the automaton is converted into a PN using a ProM plug-in. A prediction of the next event of the running process instance is returned by the algorithm as output. Finally, the method presented in [1] fits a probabilistic model (PFA) according to the event log. The framework (EM algorithm) can predict the next event of the running instance returning a binary value to determine if the next event will be of a determined type. As inputs, the method receives the sequence flow and the number of occurrences of each event type. The resulting probabilistic model can be easily understood and visualized. The software of this proposal is available.[5] An extension of this work is presented in [5]. This method determines the next activity in a running instance by applying probabilistic finite automaton. The method, named RegPFA, receives the whole process modelled as a PN, as input. Understandable predictive models are provided by this approach.

The work described in [61], provides a methodology for the representation of five different models of path followed by a BP instance, even considering the parallel execution paths. These models are trained using a DT classifier. The study analyzes the accuracy of the prediction and the complexity of the tree for each representation.

The study published in [40] performs a comparison among three types of BP monitoring predictive methods. The methods were classified according to the type of methodology employed among machine learning, constraint satisfaction techniques and quality of services (QoS) aggregation. The machine learner employed in this experimentation is the artificial neural network (ANN). Constraint satisfaction technique generates a formulation of the constraints of the problem and execute a constraint solver. These constraints consider several aspects of the process such as the event sequences, conditional executions, loops and execution times. Finally, the QoS approach, checks the QoS violations and determines a set of QoS aggregation rules for the process. The control-flow perspective is encoded as input for the algorithm and it returns a discrete value indicating if a violation will occur in the system. Authors also analyzed the improvement of the accuracy using an individual technique or a combination of the different presented techniques (ensemble learning). All data for the experimentation is available.[6]

The work described in [14], presents an approach for predictive risk monitoring (PRISM) that automatically propagates risk information, which has been detected via risk sensors, across similar running instances of the same process in real-time. This method is based on similarity-weighted process instance graphs (PING) and receives as input, the event log and predicts different risk probabilities over similar process instances.

## 3.2 Non-Process Aware Approaches

Non-process aware approaches are referred to those works whose prediction model do not use an explicit representation of the process model. These techniques and their achieved results are summarized in Table 3.

### 3.2.1 Regression Methods

Among the non-process-aware regression methods, 5 of 6 proposals are based on classical machine learning algorithms and 1 is based on statistical models (HMMs). 3/6 cases use RMSE as quality measure and [2] achieved the best results. SLA violations are predicted in 3 works, and remaining time is predicted in also 3 of the proposals. Lastly, a public dataset is only used in [58].

The following approaches incorporate regression methods for the prediction of remaining time of the processes. The proposal described in [3] converts a process instance into a set of context properties and attributes of process. A clustering method is used to select the most significant structural patterns to make the forecast. The clustering method considers the context data and target variables derived from performance values. Three different regression algorithms (Linear regression, RepTree and IB-k) are used for the prediction. The inputs of the algorithm are the traces of a log event, and a target performance measure (in this case, the remaining processing time). Some derived attributes and context information are also included in the encoding. The second approach is a process performance predictor framework, presented in [2]. As predictive approaches, this work uses regression models, pattern mining and clustering methods. This method includes a novel monitoring architecture, and incorporates information of the ongoing process such as performance statistics or notification of SLA violations. Inputs of the method are target performance measure, a threshold, a base regression method, the sequence of events, some context data of the events and some context features.

The proposal described in [33] uses a regression method for durations between two-point measures in the process (checkpoints). The predictions are then used to identify whether a service level agreement will be violated. As input data for the prediction model, the authors take into account both SLAs, process instance data and estimators (e.g., service response times). Regression methods are employed to predict values of service level objectives (SLOs) that represent the outputs of the method. In [58], authors propose the use of Long Short-Term Memory (LSTM) neural networks for different predictive monitoring tasks including remaining time

4. https://github.com/ArikSenderovich/P3Folding/
5. uni-due.de/zlv
6. s-cube-network.eu

TABLE 3
Summary of Non-Process-Aware Approaches for Regression and Classification Problems

| Author, Year (Name) | Ref | Q | Q Type | Dataset | Method | Prediction |
|---|---|---|---|---|---|---|
| Regression methods | | | | | | |
| Leitner, 2010 | [33] | 8.0 | Avg. error | - | RT | Time, SLA |
| Folino, 2012 (CA-PPM) | [23] | 0.50 | RMSE | 5,336 traces | PCT | Time, SLA |
| Bevacqua, 2013 (AA-TP) | [3] | 0.28 | RMSE | 5,336 traces | Cluster | Time |
| Bevacqua, 2014 (AA-PPM) | [2] | 0.28 | RMSE | 5,336 traces | Cluster | Indicator |
| Tax, 2017 | [58] | 71.2 | MAE | Italian sw co., BPI Ch. 2012 | LSTM NN | Time, next event |
| Classification methods | | % | | | | |
| Kang, 2011 | [28] | 10.3 | Overall error | 1,030 instances | SVM | Indicator |
| Kang, 2012 | [29] | 70.0 | Precision | 10,000 instances | KNN, LOF | Risk |
| Leitner, 2013 (E-dict) | [32] | 93.9 | Precision | 5,000 traces | DT, NN | SLA |
| Leitner, 2013b (PREVENT) | [43] | - | - | - | DT, ANN | SLA, Agg. att. |
| Pika, 2013 | [44] | 90.5 | Precision | BPI Ch. 2012 | Stat. method, PRI | Risk |
| Pika, 2013b | [45] | 80.0 | Precision | - | PRI | Risk |
| Cabanillas, 2014 | [6] | 87.8 | F-score | 119 event logs | SVM | Risk |
| Maggi, 2014 | [35] | 70.5 | Precision | BPI Ch. 2011 | DT | LTL rules |
| Conforti, 2015 | [13] | - | - | 1,065 traces | DT | Risk |
| Folino, 2015 (APP-mine) | [24] | 84.5 | Precision | 5,336 traces | Cluster, time-series | Aggregated attr. |
| Francescomarino, 2015 | [25] | 81.0 | Accuracy | BPI Ch. 2011 | Clustering, DT | LTL rules |
| Leontjeva, 2015 | [34] | $>80.0$ | AUC | BPI Ch. 2011 | HMMs, RF | Indicator |
| Verenich, 2015 | [66] | 91.2 | AUC | BPI Ch. 2011 | Cluster | Next event |
| Francescomarino, 2016 | [21] | 87.0 | Accuracy | BPI Ch. 2011 and 2015 | Clusters, optim. | LTL rules |
| Teinemaa, 2016 | [59] | 79.1 | F-score | LtC process | RF, text mining | Risk |
| Verenich, 2016 | [65] | 81.2 | AUC | Bondora and CoSeLog | SVM | Next event |
| Marquez-Chamorro, 2017 | [37] | 89.1 | F-score | BPI Ch. 2013 and SAS log | EC, decision rules | Indicator |

and next task predictions. Authors use different number of layers of the NN and different lengths of event window for the prediction.

A clustering-oriented method, presented in [23] predicts processing times and associated SLA violations. The instance is assigned to a reference scenario (cluster) which is used for the prediction. The predictive model is based on DTs and is called Predictive Clustering Tree (PCT). The definition of these clusters, generated by the Predictive Clustering submodule, can be represented as a set of logical decision rules and groups the traces according to similar target values. The inputs of the method are a log event with data attributes and environment features, a target measure and a threshold of risk.

Finally, in this section, we have also included a test-bed for the evaluation of BP prediction techniques described in [42]. This test-bed consist on an architectural framework for the simulation of BP and prediction techniques. The system is composed of three layers: semantics, core/middleware, and persistence. In the first layer, a BPMN-XPDL schema which describe a BP can be defined. The simulation of the process is carried out in the second layer which comprises four different elements: a message queue, a BP Engine (BPE), prediction algorithms and a BPM service. The persistence layer stores the events, process states and prediction data generated by the system. One of the prediction methods developed in the framework is based on HMMs and is performed to discover the forecasting time to completion of a process.

#### 3.2.2 Classification Methods

Non-process-aware classification methods represent the majority of proposals anlysed. Among them, 13 of 17 proposals are based on machine learning (6 DTs, 4 clustering methods and 3 SVMs among others) and 4 are based on statistical methods and probability models. 6 cases use precision as quality measure and F-score and AUT is used in other 3, respectively. A public dataset is used in 8 of the proposals. Finally, the software of 3 proposals can be downloaded ([65], [32] and [55]).

*Support Vector Machines*. SVM is employed as performance indicator predictors in the following two methods. In [6], authors define the monitoring of tasks as a set of requirements for a predictive system. A SVM approach is used to classify a successful completion of the process. Authors take into account the air traffic information of an airline to determine possible diversions in the landing. The attributes employed for the prediction are the geographical coordinates of the aeroplane, the covered distance and speed of the plane. The method showed in [28] periodically predicts the performance of the process and its ongoing status. This method determines possible paths of the running instance. As inputs of the SVM, authors employed the sequence flow and eight attributes of the events. Finally, to reduce the overprocessing of BP, this work [65] proposes a predictive model based on SVM. Specifically, the model aims to predict the probability that a knockout check, which classifies a case into accepted or rejected, leads to a positive value and the effort of this check in terms of processing time. As input, the approach utilizes feature vectors with the data event information. The scripts, the datasets and the results of this work are available.[7]

*Decision Trees*. Decision trees (DT) are considered in the following methods mentioned below. First, the violation of linear temporal logic constraints is predicted in [35]. Authors develop a system to estimate the probability of satisfying LTL constraints and also provides recommendations

7. http://apromore.org/platform/tools

to maximize this probability according to the event log. These constraints are defined in terms of LTL rules. An example of a LTL rule could be: $G(eventA \rightarrow F(eventB))$, where $F(x)$ indicates $x$ that is true sometimes in the future and $G(x)$ means that $x$ is true always in the future. The inputs of the algorithm are event data information in form of typed variables for each trace, and the output are the generated predictions and recommendations. The second method, proposed in [13], determines the probability of a risk in the system. This technique is based on DTs, which receives from the event log several attributes, such as resources, activity durations and frequencies. It also takes as input the current workflow (sequence of events) and the future event. The output of the method is the probability of risk in the system for the execution of this event. This method can be considered a decision support system (DSS). The DT determines the probability that a certain risk occurs in the system. This method was implemented as a YAWL plug-in. This work is an extended version of the one presented in [12]. Finally, [59] present a predictive process monitoring framework that combines text mining with sequence classification techniques so as to handle both structured and unstructured event payloads (textual information). Textual features vectors are incorporated for the encoding in order to improve the predictions. Different techniques, such as LDA and NB are considered for the extraction of text models. Random forest and logistic regression are applied for the classification stage. The method[8] receives different lengths of feature vectors and returns a probability of ocurrence.

*Clustering Methods*. Clustering methods are used in the following two proposals. In [24], authors try to determine a violation of an aggregate indicator in several checkpoints of the ongoing instance. The aggregate attributes are those calculated using several process instances over an interval of time. The algorithm, named APP-mine, is divided into three parts: a calculation of the aggregate metrics, a clustering approach for the prediction of performance model and two time-series prediction models. The inputs of the method are an aggregate PPI (A-PPI), composed of an aggregate metric and an upper threshold, the sequence of events and their data properties. Finally, in [25], authors describe a clustering and decision method for the prediction of the fulfilment or violation of a determined predicate (LTL rule). First, each running instance is assigned to a cluster according to their similarities with the historical traces and a DT is built for each cluster. This supervised machine learning method determine the prediction according to the generated model. The sequence of events and their frequency of the occurrence are taken into account in the encoding of the traces. The DT receives as input data the different clusters as training sets, as well as the constraints to predict. A binary value to determine the fulfilment or violation of a certain constraint is returned by the classifier method as single output. An extension of this work is presented in [20]. This prediction framework allows to run different configurations for different combinations of techniques.

*Hybrid Methods*. The following approaches combine several methodologies to increase the performance of their methods. The method presented in [32], allows SLA compliance prediction for running BP instances. The quality indicators for SLOs which are individual performance metrics called service-level objectives is evaluated. DT, NN and ARIMA models are employed for the prediction of nominal SLOs (e.g., order fullfilment time), metric SLOs and aggregated metric SLOs, such as the response time, respectively. Internal and external metrics and event data are used as inputs of the method. A single boolean SLO value is returned as output by the algorithm. The presented method, named E-dict tool, is based in VRESCo.[9] The same authors present the PREVENT framework (prediction and prevention based on event monitoring) in [43], which uses multilayer ANNs for the prediction of quantitative SLOs and C4.5 DTs for qualitative SLOs. In [66], a combination of clustering methods (hierarchical clustering and k-medoids) and multiple classifiers is presented. Historical traces of the processes are clustered according to their control flow information. Afterwards, each cluster is trained and a classifier model is obtained. The running instance is assigned to a determined cluster and the corresponding model is applied to obtain the prediction of the most likely value. Sequence of events with their corresponding data and different lengths of event window are considered as input of the method. Finally, [21] propose a framework to combine and tune different techniques using hyperparameter optimization in order to predict different targets, defined as LTL rules, of the running instance. The proposal consists of three steps: clustering (e.g., K-means or agglomerative clustering), classification (e.g., decision or random trees) and optimization of parameters. Different techniques are applied for each step and, regarding the results, the best combination of techniques is figured out. Control and data flow perspectives of processes, besides the frequency of occurrence of the events are considered as inputs and the method returns the compliance of a determined LTL rule.

*Other Methods*. Authors propose a set of process risk indicators (PRIs) than can be predicted using statistical techniques in [44]. These indicators (5 different types) are related to some measures such as abnormal execution time, excessive number of resources or repetition of multiple events. This method receives a sequence of activities as input. First, they analyze the event log to determine a threshold, which is used to predict possible outliers. On the other hand, a prediction function defines the degree of risk of each case using the cited risk indicators. An extension of this work is presented in [45] where authors propose a statistical technique, to estimate outliers detection according to the cited PRIs. For each indicator, they determine a threshold for the process delay prediction. The method received the sequence of activities as input extracted from two event logs.

Authors describe in [29] a real-time monitoring system which predicts the abnormal termination of a running BP. The machine learning approach employed is a K-Nearest Neighbor (KNN) technique in combination with a local outlier factor (LOF) approach as a fault detection algorithm. This factor determines how isolated is a pattern compared with the others. Distribution of LOF values are calculated and the probability of abnormal termination are also estimated during the process and returned by the algorithm.

8. https://github.com/irhete/PredictiveMonitoringWithText

9. infosys.tuwien.ac.at/prototypes/VRESCo/

TABLE 4
Summary of Methods According to the Categories Presented in the Paper: Process (PA)/Non-Process Aware (NPA), Classification (CLASS)/Regression Method (REG), Input Data Type (INPUT), Object of Prediction (PREDICT) and Methodology (METHOD)

| Author, Year (Name) | Ref | PA | NPA | CLASS | REG | INPUT | PREDICT | METHOD |
|---|---|---|---|---|---|---|---|---|
| Leitner, 2010 | [33] | | X | | X | DATA | SLO | RT |
| Aalst, 2011 (FSM Analyzer) | [63] | X | | | X | SEQ, TS | TIME | ATS |
| Kang, 2011 | [28] | | X | X | | SEQ, DATA | IND | SVM |
| Folino, 2012 (CA-PPM) | [23] | | X | X | X | SEQ, DATA, EXT | TIME, SLO | CLU, DT |
| Kang, 2012 | [29] | | X | X | | DATA | RISK | KNN, LOF |
| Metzger, 2012 (FInest) | [39] | X | | | X | SEQ | IND | STAT |
| Bevacqua, 2013 (AA-TP) | [3] | | X | | X | SEQ, DATA | TIME | CLU, RT |
| Leitner, 2013 (E-dict) | [32] | | X | X | | DATA, EXT | SLO | ANN, DT |
| Leitner, 2013b (PREVENT) | [43] | | X | X | | DATA | SLO, AGG | ANN, DT |
| Pika, 2013 | [44] | | X | X | | SEQ | RISK | STAT |
| Pika, 2013b | [45] | | X | X | | SEQ | RISK | STAT |
| Rogge-Solti, 2013 (Stoch. PN) | [48] | X | | | X | SEQ, TS, EXT | TIME | STAT, ATS |
| Becker, 2014 | [1] | X | | X | | SEQ, FREQ | NEXT | STAT |
| Bevacqua, 2014 (AA-PPM) | [2] | | X | | X | SEQ, DATA | IND | CLU, RT |
| Breuker, 2014 | [4] | X | | X | | SEQ | NEXT | STAT, EM |
| Cabanillas, 2014 | [6] | | X | X | | SEQ, DATA | RISK | SVM |
| Maggi, 2014 | [35] | | X | X | | DATA | LTL | DT |
| Metzger, 2014 | [40] | X | | X | | SEQ | RISK | ANN, AGR |
| Polato, 2014 | [46] | X | | | X | SEQ, TS | TIME | SVM, STAT |
| Conforti, 2015 | [13] | | X | X | | SEQ, DATA | RISK | DT |
| Folino, 2015 (APP-mine) | [24] | | X | X | | SEQ | AGG | CLU, TS |
| Francescomarino, 2015 | [25] | | X | X | | SEQ, FREQ | LTL | CLU, DT |
| Lakshmanan, 2015 | [31] | X | | | X | SEQ | NEXT | HMM |
| Leontjeva, 2015 | [34] | | X | X | | SEQ, DATA | IND | HMM, DT |
| Rogge-Solti, 2015 | [49] | X | | | X | SEQ, TS, FREQ | TIME | ATS |
| Senderovich, 2015 | [55] | X | | | X | SEQ, DATA, TS | TIME | QT, ATS |
| Verenich, 2015 | [66] | | X | X | | SEQ, DATA | NEXT | CLU, DT |
| Breuker, 2016 RegPFA | [5] | X | | X | | SEQ, FREQ | NEXT | SVM, STAT |
| Cesario, 2016 | [10] | X | | | X | SEQ, FREQ | TIME | CLU, ATS |
| Conforti, 2016 | [14] | X | | X | | SEQ | RISK | SIM |
| Francescomarino, 2016 | [21] | | X | X | | DATA, FREQ | LTL | CLU, DT |
| Polato, 2016 | [47] | X | | | X | SEQ, DATA | TIME | SVM, STAT |
| Senderovich, 2016 (P3-FOLD) | [53] | X | | | X | SEQ, FREQ | TIME | ATS |
| Song, 2016 | [60] | X | | | X | SEQ, FREQ | IND | ATS |
| Teinemaa, 2016 | [59] | | X | X | | DATA, FREQ | RISK | STAT, DT |
| Unuvar, 2016 | [61] | X | | X | | SEQ | NEXT | DT |
| Verenich, 2016 | [65] | | X | X | X | DATA | NEXT, TIME | SVM |
| Marquez-Chamorro, 2017 | [37] | | X | X | | DATA, EXT | IND | EC |
| Tax, 2017 | [58] | | X | X | X | DATA | NEXT, TIME | ANN |

*The acronyms for input data type: control-flow and time perspectives (SEQ), data-flow perspective (DAT), External attributes (EXT), the frequency of events (FREQ) and a state transition system (TS). The acronyms for objects of prediction (PREDICT): remaining times and delays (TIME), SLO predictions (SLO), risk Prediction (RISK), indicator value (IND), LTL rules (LTL), aggregate metrics (AGG) and next event (NEXT). The acronyms for the different methodologies are: SVM (support vector machine), KNN (k-nearest neighbor), STAT (statistical techniques), SIM (similarity measures), QT (queueing theory), TS (time series), HMM (hidden Markov model), ANN (artificial neural network), DT (decision tree), CLU (clustering method), RT (regression tree), LOF (local outlier factor), EM (expectation maximization), AGR (QoS aggregation rules), ATS (annotated transition and graph-based systems) and EC (Evolutionary computation).*

The algorithm receives a dataset based on 10,000 generated instances composed of 6 relevant process attributes.

Recently, the first evolutionary computation (EC) approach for predictive monitoring has been presented in [37]. This evolutionary method applies an event window-based encoding and generates a set of decision rules, easily understandable by the user, for the prediction of some indicator values. Furthermore, a full software stack for the training phase and a framework for the integration of run-time predictions with BP management systems, has also been developed.

In [34], different encodings are generated for the prediction. These encodings, used as inputs of the method, are represented by feature vectors which includes the sequence of events and some information such as the order of the events, their frequencies of appearance or the attribute values of the last event. A random forest classifier is applied to the resulting models and returned a binary value to determine whether a temporal constraint is fulfilled or not for a determined instance. Best results were achieved using the HMM encoding.

## 4 Discussion

This section is divided according to the different concepts and stages of the methodology described in Section 2. Table 4 shows a compilation of all the methods categorised by the different classifications described in Sections 3 and 4: Process/Non-process aware, Classification/Regression method, Input data type, Object of prediction and Methodology.

### 4.1 Input Data and Data Sets

Input data of the predictive algorithms is considered for the classification showed in Table 4. We have taken into account

TABLE 5
Summary of Data Sets Employed for the Experimentation of the Different Predictive Methods

| Name | Size | Refs | Availability |
|---|---|---|---|
| <<unknown>> | 1,000 traces | [33] | n/a |
| Dutch municipality | 796 cases | [63] | n/a |
| Manufacturing process | 1,030 instances | [28] | n/a |
| Transshipment system | 5,336 traces | [2], [3], [10], [23], [24] | n/a |
| Logistic provider | 10,000 traces, 784 cases | [29], [48] | n/a |
| CARGO 2000 system | 3,942 instances | [39], [40] | yes |
| ACMEBOT process | 5,000 traces | [32] | n/a |
| BPI Challenge 2011 (Dutch Academic Hospital) | 1,100 cases | [21], [25], [34], [35], [66] | yes |
| BPI Challenge 2012 (Dutch financial bank) | 13,087 traces | [4], [5], [44], [49], [58] | yes |
| BPI Challenge 2015 (Dutch municipalities) | 1,199 cases | [20], [21] | yes |
| BPI Challenge 2013 (Volvo IT incidents) | - | [5], [37] | yes |
| <<unknown>> | 3,777 instances | [1] | n/a |
| Air traffic information | 119 event logs | [6] | n/a |
| Sending for credit collection | 1,500 and 5,000 traces | [46] | n/a |
| Insurance company (claim handling process) | 1,065 traces | [13] | n/a |
| Israeli bank call center | 7,000 traces | [55] | n/a |
| Road fines log | 7,300 traces | [47] | n/a |
| US Hospital | - | [53] | n/a |
| Personal loan process | 9,350 instances | [14] | n/a |
| Marketing campaign process | - | [61] | n/a |
| LtC process | - | [59] | n/a |
| Italian software company | 3,804 cases | [58] | n/a |
| Bondora and CoSeLog | 40,062 and 1,230 | [65] | yes |

the different process perspectives described in Section 2.3, such as the control-flow and the data-flow perspective. External and computed attributes have also been considered for the classification. Other aspects as the frequency of events is also taken into consideration as input data. We can appreciate that the majority of the methods use the sequence of events for the prediction. Data-flow perspective is also considered in approximately half of the cases. Frequency of events is used in a 20 percent of the analyzed methods. External attributes and contextual information (e.g., the weather) and state transition systems which provides a complete model of the process, are considered in a 14 and 11 percent of studied cases, respectively. The main conclusion is that the event data adds valuable information to the predictive method and is becoming in an essential input. Consequently, the higher the number of event data attributes, the higher computational time and CPU consumption is required by the algorithm. Thus, feature selection is necessary to determine which are the most valuable attributes for the predictive process in each case.

Data sets used in the experimentation of the methods are collected in Table 5. First column indicates the name of the event log. Second column shows the size of this data set, e.g., number of traces or number of process instances. Third column presents the reference of the paper where the data set is used. Finally, fourth column indicates the availability of the data set. Business Process Intelligence Challenge (BPIC)[10] provides public data sets every year for this competition. The rest of datasets belong to company cases and data are generally not available, which limits the ability to compare between techniques. Furthermore, even if the proposals use a publicly available dataset, the comparison between approaches may be difficult because because the object of prediction, i.e., the way the predicted value is computed from the event log, is not precisely defined.

### 4.2 Encoding

Although several of the proposals are not explicit enough on this point, encodings are generally based on feature vectors which encode the sequence of events and event data information, e.g., [24]. This is particularly common in most approaches that rely on well-known machine learning algorithms. Other methods include the whole process model in the encoding, such as PNs used in [63]. Some proposals also enrich the information of the event logs with external attributes, such as [48], or with metrics computed from the event log itself [16]. Furthermore, the value of the metric to be predicted should be computed for each trace. In [36], several indicators are calculated using a ProM plugin.

With respect to the determination of the best encoding, a cited paper must be the focus of our attention [34]. In this paper, authors performed different encodings for the input data, trying to find which is the optimal one. As conclusion, the encoding based on HMMs obtains a slight advantage in performance accuracy terms with respect to others. Other aspect to be considered in the encoding is the historic of events.

A higher number of data events for the encoding implies a decrease in the efficiency of the system. To solve this problem, a variation in the number of events for the prediction is considered in several works, e.g., [5], [37]. Authors in [34], propose an encoding using only the data associated to the last event to deal with this problem.

### 4.3 Building the Model

According to the process awareness of the methods (Table 4), we conclude that 23 of 39 proposals do not utilise a process model in their methodologies against a total of 16

10. win.tue.nl/bpi/doku.php?id=2016:start_previous_events

methods which incorporate a process model for the prediction. Furthermore, those that include a process model for the prediction tend to be used to predict either time (8/16) or the next activity in the process (5/16). Instead, there is no clear tendency concerning non-process-aware models.

It is also significant the relationship between process awareness and the type of predicted value. Amongst process-aware methods, there are 9 of 16 proposals that focus on predicting a numeric value (regression), whereas only 7 of 23 non-process-aware methods focus on that problem. Therefore, we can conclude that process-aware proposals tend to be used for regression, whereas non-process-aware methods are more used for classification.

Attending the scope of the prediction, the majority of methods (36/39) focus on predictions of a single process instance, whereas just two proposals focus on predictions of aggregate metrics. Furthermore, since the nature of the prediction is different, the methods used in these approaches differ from those used to make predictions of a single process instance. For instance, [24] is the only paper that includes the use of time series in the predictions.

Considering the type of methodology, we can conclude that the majority of the predictive proposals (28/39) include a machine learning approach (i.e., SVM, KNN, ANN, DT, CLU and RT). Among this type of techniques, DTs were the most used approaches, 11/28, followed by clustering approaches (8/28) and SVM approaches (6/28). Statistical techniques, such as Naive-Bayes or probabilistic methods, rank second with 10/39. Finally, annotated transition and graph-based systems were used as a predictive model in 8/39 cases.

Another conclusion is that a significant number of approaches (20/39) combine several techniques to build the prediction model. The most common combination is to include a first part where the instances are grouped according to their similarities, e.g., clustering method, and a second classification part where a machine learning approach, such a DT, determine the predicted value of a process indicator. These combinations of methods seems to improve the performance rate of the proposals according to the quality assessment values of Table 3.

Other relevant aspect is the selection of checkpoints. Some of the methods, such as [24], [40] and [33], apply this concept in their methodologies.

Finally, the interpretability of the model obtained has not been a main concern of the proposals included in this survey. Only a few proposals have explicitly mentioned the interpretability of the model as a relevant factor when choosing the predictive model: in [5] authors provide a useful design to visualize the probabilistic models, to determine the behavior of a process instance in the future. Recently, in [37], an evolutionary approach generates a set of decision rules for the run-time prediction of process indicators that can be easily interpreted by users to extract further insight of the BPs and [1] returns a probabilistic model that can be easily understood and visualized. However, given the difficulty to compare the proposals, it is not possible to conclude whether there is a trade-off between the interpretability of the model and its performance. In other words, it is not possible to conclude whether interpretable models such as DTs or some annotated transition systems perform better (or worse) than harder-to-understand models such as SVM or ANN models. An additional problem is that in real-world event logs in which many categorical variables appear (e.g., product name or resource), it is difficult to obtain useful insights even in models that are easier to understand such as DTs. The reason is the complexity of these models grows significantly with these kind of variables.

TABLE 6
Summary of Prediction Methods According to the Different Application Domains

| Type of predicted values | References |
|---|---|
| Remaining time | [3], [23], [42], [46], [48], [49], [58], [63] [10], [47], [53], [55], [61], [65], [66] |
| Risk probability | [6], [13], [14], [29], [40], [44], [45], [59] |
| Any value of indicator | [2], [3], [20], [28], [32], [33], [34], [39] [37], [60] |
| LTL rule | [21], [25], [35] |
| Aggregate metrics | [24], [32] |
| Next event | [1], [4], [5], [47], [58], [61], [65], [66] [31] |

### 4.4 Predictions

Existing techniques can also be organized according to the object of prediction of the proposals (Table 6). These predicted values are time, mainly remaining execution time, the risk probability (e.g., the violation of a constraint), a LTL formula which determines that a certain situation in the process occurs, an estimation of the value of a single indicator or an aggregate attribute and, finally, the prediction of the next event of the running process instance. Methods that estimate the remaining time and delays in the execution rank first with 15/39 proposals. Second, 10/39 methods that predict a single attribute. Those methods which predicts the probability of a certain risk, such as an abnormal termination, and the next event of the process rank third with 8/39 proposals each one.

From the analysis of the collected works, we conclude that classification methods are used to predict SLO values, risk indicators and next events while regression methods generally predict time. Other types of indicators are predicted by both type of methods, depending on whether they are predicting a specific value of the indicator or whether the indicator is going to be met or not. Aggregated metrics are also predicted by both types of methods.

Finally, although the comparison between different types of predictions is not possible, we can observe that there is not significative differences in the quality of the results obtained depending on the object of prediction.

### 4.5 Evaluation of the Model

Two main conclusions can be extracted from the previous study concerning the evaluation of the proposals. The first of them is the lack of an exhaustive comparison in the presentation of the works. The different data sets, quality metrics and input features employed, hinder the comparison. Although the majority of methods achieve a reasonable prediction performance, i.e., precision and accuracy rates higher than 70 percent, the lack of comparison prevents the determination of which of the proposals obtain the best global performance rates. The second of them is the absence of available software of the proposals, and in consequence, users are not allowed to test the validity of these methods using different data sets. However, published works in latest years are reversing this trend and their software can be downloaded (e.g., [59], [65]). Finally, regarding the type of evaluation measures, we can conclude that most of regression methods employ RMSE for the evaluation (8/16) and the most utilised measures for classification are Accuracy (14/22) and Precision (7/22).

## 5 Conclusions and Challenges

Event logs provided by information systems records all data about process executions, and this information is useful for building models that enable a predictive BP monitoring. Considering these predictions, we can anticipate the occurrence of problems so they can be prevented, managed and mitigated. In this context, we present a compilation of a total of 39 works for predictive monitoring of BP based on different predictive techniques. These methods have been classified according to the process-awareness of methods and the technique used to build the predictive model (classification or regression). Several features as the quality of prediction, the data sets employed, and the inputs and outputs of the method have been also considered. Comparing the performance of the different approaches, we cannot draw clear conclusions to determine which is the best methodology. It depends on the data set used, the input features of the machine learning algorithm, among other issues.

Concerning the evolution of the different proposal over the years, we have considered several highly-cited works previous to 2010. In [57], authors implements a BPI methodology for the prediction of general metrics or metrics defined by the user. In [9], authors used class-based time series to generate a model for the predictions and decision makings through a GUI providing explanations and predictions to the users. In [27], authors implement a BPI tool for the analysis and prediction of exceptions. They define and analyse these exceptions from the event log, and obtain a DT for the prediction. Finally, authors in [56], define an architecture for the evaluation of the risks and the prediction is carried out or decision making module. We can appreciate that a detailed experimentation is not provided in these cases, so an exhaustive comparison is not possible with the works presented in this paper (from 2010 to 2016). Other conclusions concerning the evolution of the methods over the years can be, considering the different techniques employed, from 2010 to 2013, non-process aware methods predominated. From 2014 until now, where the majority of the papers are published, the number of classification methods overcomes the number of regression methods, and the number of process and non-process aware methods are balanced.

Although this work has been mainly focused in BPs, predictive monitoring can also be applied to other types of industrial processes such as production, manufacturing or case handling processes. Some of them has been collected in Section 3 ([39], [48], [60]). Recently in [60], the cost of a manufacturing process is analysed and predicted. A process model-enhanced cost, as well as predictions of volume and remaining times, are used as process mining techniques in this case. Finally, in [67], adaptive neural networks and a nonlinear model predictive control (NMPC), assist a industrial process control for the optimization of a performance index avoiding delays and package dropouts.

Some challenges related to predictive monitoring can be identified as a result of this study. Next, we detail them grouped in 5 broad categories.

### 5.1 Application of New Technologies

Machine learning methodologies, on which most proposals rely, require large computing and processing capabilities for the management and analysis of their vast historical data sets stored by information systems. Therefore, software for big data could play an important role to solve this issue. Some works have started to deal with this challenge. The methodology proposed in [15], combines the well-known process mining framework ProM with the platform Apache Hadoop[11] for the distributed processing of large event data sets.

Another issue is the processing of real time event streams. The high volume and speed of these continuous flows of data represents a great challenge for building and maintaining predictive models. Spark streaming environment[12] can face some of these problems. This architecture can build scalable streaming applications, consume static and streaming data from various sources and also apply machine learning techniques.

At last, a new open challenge related with the new technologies in BPM is introduced with the incipient use of blockchains [38]. Blockchain is a distributed database for decentralized and transactional data sharing across a network of participants, where blocks of information are recorded and linked between them to facilitate the retrieval of information and the verification of changes. Considering the predictive monitoring, we will address some issues such as the data fragmentation and encryption of data. The source of data, typically a single event log, will be replaced by multi-sources of events. An adaptation of BPM systems with the addition of new solutions and software components will be needed. In [68], authors propose a combination of components, such as a transaction history and smart contracts, for the execution and monitoring of BPs with blockchains.

### 5.2 Evaluation

The analysis of the collected works has revealed a lack of comparison among the different proposals that make it difficult to tell which is the most appropriate for each situation. Besides the fact that many datasets used are not public and that the software is generally not available, it is necessary to provide the community with a workbench that makes the comparison between different proposals easier and replicable. This includes mechanisms to define precisely the prediction that is being made, to detail which quality metrics are used in the comparison and to characterise the datasets.

### 5.3 Objects of Prediction

Many of the process performance indicators used by organizations nowadays are aggregation of measures over an interval (e.g., the percentage of incidents solved in time in a month). However only a couple of proposals [24], [32] deal with them. Predicting these measures raises new interesting

11. hadoop.apache.org
12. spark.apache.org/streaming

challenges since, in order to predict them, one needs to predict how many future instances there will be in the remainder of the aggregation interval.

Another interesting aspect would be to understand whether there are some differences in the performance of the prediction depending on the domain to which the prediction is applied (performance indicators, risk, SLA violations, and other values). Understanding these differences would help to develop techniques specially tailored for one domain or, at least, to select the most appropriate technique in each moment.

### 5.4 Quality of Predictions

There are several lines of work that could be followed to improve the quality of the predictions. First, most proposals usually feed the learning algorithm with the information from the event log as is. However, little work has been done concerning the use of domain knowledge about a process event log to build new features that improve the prediction power of models, or the selection of key attributes in order to reduce the computational time and CPU consumption in a monitoring prediction task. This is specially relevant if the log is enriched with external and computed attributes in order to increase its predictive power because the number of attributes used as inputs for the model can grow exponentially in these cases. An analysis of feature selection techniques applied in BP intelligence area is described in [26].

Second, as it is referred in [34], the majority of the presented predictive methods are focused on the intra-case predictive monitoring. They only consider the predicted value of an individual process instance, and each of them is seen independently from each other. Sometimes, an inter-case-predictive monitoring could be useful for predictions of the total ongoing process instances, e.g., percentage of active instances that ends with an abnormal termination. Some aspects of the process, such as resource contention and data sharing among processes, can be relevant in these cases. An inter-case predictive monitoring proposal is described in [13].

### 5.5 Strategy for Building and Updating Models

Another challenge that has often been neglected by the literature is the strategy to define the checkpoints and to decide when to update the predictive model. Concerning the former, it is necessary to find an equilibrium between the number of selected checkpoints and the monitoring cost. In [69], some proposals are defined to deal with this problem. However, more work is required to determine which strategy is more convenient in each case. Regarding the update of the predictive model, it is common that processes and their performance evolve with time. In this case, it becomes necessary to retrain the predictive model from time to time to ensure that its quality does not deteriorate with this evolution. Therefore, strategies must be developed to help decide when to retrain the model and which is the appropriate event window that should be considered while retraining it to optimize the performance of the model.

### 5.6 Predictions in Practice

Most proposals are focused on improving the accuracy of predictions, but little attention has been given to providing recommendations and explaining the prediction values to the users so that they can determine the best way to act upon. This may hinder significantly the applicability of predictive monitoring in real settings. Three lines have been considered related to the applicability of the proposals.

The first one is related to the interpretability of the predictive models, which have been mentioned explicitly by only a few proposals [1], [5], [37] as a concern while building predictive models. On the one hand, experiments should be conducted to understand whether there is a trade-off between the interpretability of the model and its performance. On the other hand, with real-world event logs, it is difficult to obtain useful insights even using predictive models that are potentially understandable by users such as DTs because of their complexity. Therefore, it is necessary to develop tools that help users to query these models in order to get information that is relevant for them.

The second one is related to the recommendations that can be made to the user based on the predictions. Only a couple of proposals focus on them: [13] defines a recommendation system that identifies the best assignment of resources for the current process instance based on the generated risk predictions, and [35] also provides recommendations to maximize the probability of satisfying a particular constraint. Also [50] describe a system for operational decision support based on simulations. However, there is still work to be done in this direction. For instance, the recommendations presented to the user must make sense in the domain, which means that domain knowledge have to be included to identify all potential recommendations. Another relevant aspect is how to evaluate the usefulness of recommendations in real settings. A/B testing could provide a way to approach this problem.

Currently, only a few of frameworks facilitates the integration of many different predictive monitoring techniques with BPMS. In the following, we described some works aligned with this issue. In [42], an architectural framework for the simulation of BP and prediction techniques is defined. In [37], authors describe a software stack to support all the stages of the predictive monitoring process (preprocessing, training and prediction) and can be integrated with Camunda BPMS.[13] A ProM operational support tool for predictive monitoring is also outlined in [22]. [13] provides a recommendation system that is integrated in YAWL. A business operation management platform (Enterprise Cockpit) [8] was developed over a decade ago, and provided some predictions of process instances. Finally, two recently released prototypes (Nirdizati[14] and XES Tensorflow[15]) have also been developed.

Finally, an interesting line of future work related to the applicability of predictive monitoring is to carry out a survey on different organizations that use BPM to know if they use and how they use predictive monitoring, e.g., a case health service developed by IBM Research described in [30] and some companies, such as Dell, Betfair or BT, which use Oracle Real-Time Decision framework.[16] We can also analyse the current BPMS to see if they contain predictive

13. https://camunda.org/
14. http://nirdizati.org
15. http://joerg.evermann.ca/software.html
16. http://www.oracle.com/us/products/middleware/bus-int/rtd-product-review-1885532.pdf

monitoring support, such as the novel released machine learning component in Bizagi 11.1.[17]

## ACKNOWLEDGMENTS

This work has received funding from the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes: grants TIN2015-70560-R (BELI), P12–TIC-1867 (COPAS) and Juan de la Cierva (JCF 2015).

## REFERENCES

[1] J. Becker, D. Breuker, P. Delfmann, and M. Matzner, "Designing and implementing a framework for event-based predictive modelling of BPs," in *Proc. Int. Workshop Enterprise Modelling Inf. Syst. Archit.*, pp. 71–84, 2014.

[2] A. Bevacqua, M. Carnuccio, F. Folino, M. Guarascio, and L. Pontieri, "A data-driven prediction framework for analyzing and monitoring BP performances," *Enterprise Inf. Syst.*, vol. 190, pp. 100–117, 2015.

[3] A. Bevacqua, M. Carnuccio, F. Folino, M. Guarascio, and L. Pontieri, "A data-adaptive trace abstraction approach to the prediction of business process performances," in *Proc. Int. Conf. Enterprise Inf. Syst.*, 2013, pp. 56–65.

[4] D. Breuker, P. Delfmann, M. Matzner, and J. Becker, "Designing and evaluating an interpretable predictive modeling technique for BPs," in *Proc. BPM Workshops*, 2014, pp. 541–553.

[5] D. Breuker, M. Matzner, P. Delfmann, and J. Becker, "Comprehensible predictive models for BPs," *MIS Quart.*, vol. 40, no. 4, pp. 1009–1034, 2016.

[6] C. Cabanillas, C. D. Ciccio, J. Mendling, and A. Baumgrass, "Predictive task monitoring for BPs," in *Proc. Int. Conf. Bus. Process Manage.*, 2014, pp. 424–432.

[7] C. Cabanillas, M. Resinas, and A. Ruiz-Cortés, "Defining and analysing resource assignments in business processes with RAL," in *Proc. Int. Conf. Service-Oriented Comput.*, 2011, pp. 477–486.

[8] F. Casati, M. Castellanos, and M. Shan, *Enterprise Cockpit for Business Operation Manage. Int. Conf. on Conceptual Modeling (ER 2004)*, pp. 825–827, 2004.

[9] M. Castellanos, F. Casati, U. Dayal, and M.-C. Shan, "A comprehensive and automated approach to intelligent BPs execution analysis," *Distrib. Parallel Databases*, vol. 16, no. 3, pp. 239–273, 2004.

[10] E. Cesario, F. Folino, M. Guarascio, and L. Pontieri, "A cloud-based prediction framework for analyzing BP performances," in *Proc. Int. Conf. Availability Rel. Secur.*, 2016, pp. 63–80.

[11] M. Cho, et al., "A new framework for defining realistic SLAs: An evidence-based approach," in *Business Process Management Forum*. Berlin, Germany: Springer, 2017, pp. 19–35.

[12] R. Conforti, M. de Leoni, M. La Rosa, and W. van der Aalst, "Supporting risk-informed decisions during BP execution," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2013, pp. 116–132.

[13] R. Conforti, M. de Leoni, M. L. Rosa, W. van der Aalst, and A. ter Hofstede, "A recommendation system for predicting risks across multiple BP instances," *Decision Support Syst.*, vol. 69, pp. 1–19, 2015.

[14] R. Conforti, S. Fink, J. Manderscheid, and M. Röglinger, "PRISM—A predictive risk monitoring approach for BPs," in *Proc. Int. Conf. Bus. Process Manage.*, 2016, pp. 383–400.

[15] S. H. de la Mesa, S. V. Zelst, J. E. Mateo, and W. V. der Aalst, "Handling big(ger) logs: Connecting prom 6 to Apache Hadoop," in *Proc. CEUR Workshop*, 2015, pp. 80–84.

[16] M. de Leoni, W. M. van der Aalst, and M. Dees, "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs," *Inf. Syst.*, vol. 56, pp. 235–257, 2016.

[17] A. del Río-Ortega, F. García, M. Resinas, E. Weber, F. Ruiz, and A. Ruiz-Cortés, "Enriching decision making with data-based thresholds of process-related KPIs," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2017, pp. 193–209.

[18] A. del Río-Ortega, M. Resinas, C. Cabanillas, and A. Ruiz-Cortés, "On the definition and design-time analysis of process performance indicators," *Inf. Syst.*, vol. 38, no. 4, pp. 470–490, 2013.

17. http://blog.bizagi.com/2017/06/05/artificial-intelligence-ai-bizagi/

[19] A. del-Río-Ortega, M. Resinas, and A. Ruiz-Cortés, "Defining process performance indicators: An ontological approach," in *Proc. OTM Confederated Int. Conf. "On the Move to Meaningful Internet Syst.*, 2010, pp. 555–572.

[20] C. Di Francescomarino, M. Dumas, M. Federici, C. Ghidina, F. Maggi, and W. Rizzi, "Predictive BP monitoring framework with hyperparameter optimization," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2016, pp. 361–376.

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

[22] M. Federici, et al., "A ProM operational support provider for predictive monitoring of BPs," in *Proc. Int. Conf. Bus. Process Manage. Demos*, 2015, pp. 1–5.

[23] F. Folino, M. Guarascio, and L. Pontieri, "Discovering context-aware models for predicting BP performances," in *Proc. Int. Conf. on Cooperative Inf. Syst.*, 2012, pp. 287–304.

[24] F. Folino, M. Guarascio, and L. Pontieri, "A prediction framework for proactively monitoring aggregate process-performance indicators," in *Proc. IEEE Int. Enterprise Distrib. Object Comput. Conf.*, 2015, pp. 128–133.

[25] C. D. Francescomarino, M. Dumas, F. M. Maggi, and I. Teinemaa, "Clustering-based predictive process monitoring," *IEEE Trans. Services Comput.*, 2016.

[26] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M. Shan, "BP intelligence," *Comput. Ind.*, vol. 53, no. 3, pp. 321–343, 2004.

[27] D. Grigori, F. Casati, U. Dayal, and M.-C. Shan, "Improving BP quality through exception understanding, prediction, and prevention," in *Proc. Int. Conf. Very Large Data Bases*, 2001, pp. 159–168.

[28] B. Kang, D. Kim, and S.-H. Kang, "Periodic performance prediction for real-time BP monitoring," *Ind. Manage. Data Syst.*, vol. 112, no. 1, pp. 4–23, 2011.

[29] B. Kang, D. Kim, and S.-H. Kang, "Real-time BP monitoring method for prediction of abnormal termination using KNNI-based LOF prediction," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 6061–6068, 2012.

[30] G. Lakshmanan, N. Mukhi, R. Khalaf, A. Martens, and S. Rozsnyai, "Assessing the health of case-oriented semi-structured BPs," in *Proc. Conf. Services Comput.*, 2012, pp. 499–506.

[31] G. Lakshmanan, D. Shamsi, Y. Doganata, M. Unuvar, and R. Khalaf, "A Markov prediction model for data-driven semi-structured BPs," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 97–126, 2015.

[32] P. Leitner, J. Ferner, W. Hummer, and S. Dustdar, "Data-driven and automated prediction of service level agreement violations in service compositions," *Distrib. Parallel Databases*, vol. 31, no. 3, pp. 447–470, 2013.

[33] P. Leitner, B. Wetzstein, F. Rosenberg, A. Michlmayr, S. Dustdar, and F. Leymann, "Runtime prediction of service level agreement violations for composite services," in *Proc. Service-Oriented Comput. Workshops*, 2010, pp. 176–186.

[34] A. Leontjeva, R. Conforti, C. D. Francescomarino, M. Dumas, and F. M. Maggi, "Complex symbolic sequence encodings for predictive monitoring of BPs," in *Proc. Int. Conf. Bus. Process Manage.*, 2015, pp. 297–313.

[35] F. M. Maggi, C. Di Francescomarino, M. Dumas, and C. Ghidini, "Predictive monitoring of BPs," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2014, pp. 457–472.

[36] A. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés, "Towards a general architecture for predictive monitoring of BPs," in *Jornadas de Ciencia e Ingeniera de Servicios*, pp. 1–4, 2016.

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

[38] J. Mendling, I. Weber, W. van der Aalst, J. vom Brocke, and C. Cabanillas, "Blockchains for BP Management - challenges and opportunities," arXiv:1704.03610, 2017.

[39] A. Metzger, R. Franklin, and Y. Engel, "Predictive monitoring of heterogeneous service-oriented business networks: The transport and logistics case," in *Proc. Annu. SRII Global Conf.*, 2012, pp. 313–322.

[40] A. Metzger, et al., "Comparing and combining predictive BP monitoring techniques," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 45, no. 2, pp. 276–290, Feb. 2015.

[41] H. Nguyen, M. Dumas, M. L. Rosa, F. Maggi, and S. Suriadi, "BP deviance mining: Review and evaluation," *arXiv:1608.08252*, 2016.

[42] S. Pandey, S. Nepal, and S. Chen, "A test-bed for the evaluation of BP prediction techniques," in *Proc. Int. Conf. Collaborative Comput.: Netw. Appl. Worksharing*, 2012, pp. 382–391.
[43] W. Hummer, P. Leitner, and S. Dustdar, "Cost-based optimization of service compositions," *IEEE Trans. Services Comput.*, vol. 6, no. 2, pp. 239–251, Apr.–Jun. 2013.
[44] A. Pika, W. van der Aalst, C. Fidge, A. ter Hofstede, and M. Wynn, "Predicting deadline transgressions using event logs," in *Proc. Int. Conf. Bus. Process Manage. Workshops*, 2013, pp. 211–216.
[45] A. Pika, W. van der Aalst, C. Fidge, A. ter Hofstede, and M. Wynn, "Profiling event logs to configure risk indicators for process delays," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2013, pp. 465–481.
[46] M. Polato, A. Sperduti, A. Burattin, and M. de Leoni, "Data-aware remaining time prediction of BP instances," in *Proc. Int. Joint Conf. Neural Netw.*, 2014, pp. 816–823.
[47] M. Polato, A. Sperduti, A. Burattin, and M. de Leoni, "Time and activity sequence prediction of BP instances," *arXiv:1602.07566*, 2016.
[48] A. Rogge-Solti and M. Weske, "Prediction of remaining service execution time using stochastic Petri nets with arbitrary firing delays," in *Proc. Int. Conf. Service-Oriented Comput.*, 2013, pp. 389–403.
[49] A. Rogge-Solti and M. Weske, "Prediction of BP durations using non-Markovian stochastic Petri nets," *Inf. Syst.*, vol. 54, pp. 1–14, 2015.
[50] A. Rozinat, M. T. Wynn, W. M. van der Aalst, A. H. ter Hofstede, and C. J. Fidge, "Workflow simulation for operational decision support," *Data Knowl. Eng.*, vol. 68, no. 9, pp. 834–850, 2009.
[51] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Comput. Surveys*, vol. 42, no. 3, pp. 1–42, 2010.
[52] A. Senderovich, S. Leemans, S. Harel, A. Gal, A. Mandelbaum, and W. van der Aalst, "Discovering queues from event logs with varying levels of information," in *Proc. Int. Conf. Bus. Process Manage. Workshops*, 2015, pp. 154–166.
[53] A. Senderovich, A. Shleyfman, M. Weidlich, A. Gal, and A. Mandelbaum, "P3-folder: Optimal model simplification for improving accuracy in process performance prediction," in *Proc. Int. Conf. Bus. Process Manage.*, 2016, pp. 419–435.
[54] A. Senderovich, M. Weidlich, A. Gal, and A. Mandelbaum, "Queue mining - predicting delays in service processes," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2014, pp. 42–57.
[55] A. Senderovich, M. Weidlich, A. Gal, and A. Mandelbaum, "Queue mining for delay prediction in multi-class service processes," *Inf. Syst.*, vol. 53, pp. 278–295, 2015.
[56] A. Seufert and J. Schiefer, "Enhanced business intelligence - supporting BPs with real-time business analytics," in *Proc. Workshop Database Expert Syst. Appl.*, 2005, pp. 919–925.
[57] W. Tan, W. Shen, L. Xu, B. Zhou, and L. Li, "A BP intelligence system for enterprise process performance management," *IEEE Trans. Syst. Man Cybern. Part C*, vol. 38, no. 6, pp. 745–756, Nov. 2008.
[58] N. Tax, I. Verenich, M. La Rosa, and M. Dumas, "Predictive BP monitoring with LSTM neural networks," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2017, pp. 477–492.
[59] I. Teinemaa, M. Dumas, F. M. Maggi, and C. Di Francescomarino, "Predictive BP monitoring with structured and unstructured data," in *Proc. Int. Conf. Bus. Process Manage.*, 2016, pp. 401–417.
[60] T. B. H. Tu and M. Song, "Analysis and prediction cost of manufacturing process based on process mining," in *Proc. Int. Conf. Ind. Eng. Manage. Sci. Appl.*, 2016, pp. 1–5.
[61] M. Unuvar, G. T. Lakshmanan, and Y. N. Doganata, "Leveraging path information to generate predictions for parallel BPs," *Knowl. Inf. Syst.*, vol. 47, no. 2, pp. 433–461, 2016.
[62] W. van der Aalst, *Process Mining: Data Science in Action*. Berlin, Germany: Springer, 2016.
[63] W. van der Aalst, M. Schonenberg, and M. Song, "Time prediction based on process mining," *Inf. Syst.*, vol. 36, no. 2, pp. 450–475, 2011.
[64] B. van Dongen, R. Crooy, and W. van der Aalst, "Cycle time prediction: When will this case finally be finished?" in *Proc. OTM Confederated Int. Conf. "On Move Meaningful Internet Syst."*, 2008, pp. 319–336.
[65] I. Verenich, M. Dumas, M. La Rosa, F. M. Maggi, and C. Di Francescomarino , "Minimizing overprocessing waste in BPs via predictive activity ordering," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2016, pp. 218–229.
[66] 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. Workshop Bus. Process Intell.*, 2015, pp. 218–229.
[67] T. Wang, H. Gao, and J. Qiu, "A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 416–425, Feb. 2016.
[68] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Untrusted BP monitoring and execution using blockchain," in *Proc. Int. Conf. Bus. Process Manage.*, 2016, pp. 329–347.
[69] D. Cao, X. Liu, Y. Yang, and D. Yuan, "Selecting checkpoints along the time line: A novel temporal checkpoint selection strategy for monitoring a batch of parallel BPs," in *Proc. Int. Conf. Softw. Eng.*, 2013, pp. 1281–1284.

**Alfonso Eduardo Márquez-Chamorro** is a postdoctoral researcher with the University of Sevilla, Spain, and a member of the ISA Research Group. His current research interests include machine learning, business process management, and process mining. Previously, he worked on bioinformatics and evolutionary computation.

**Manuel Resinas** is an associate professor with the University of Sevilla, Spain, and a member of the ISA Research Group. His current research interests include analysis and management of service level agreements, business process management, process performance analytics, and cloud-based enterprise systems. Previously, he worked on automated negotiation of service level agreements.

**Antonio Ruiz-Cortés** is a full professor of software and service engineering and he heads the Applied Software Engineering Group, University of Sevilla. His current research focuses on service-oriented computing, business process management, testing and software product lines, being the recipient of the Most Influential Paper of SPLC 2017 award. He is an associate editor of the *Springer Computing*.