IT Incident Solving Domain Experiment on Business Process Failure Prediction

Pedro O. T. Mello[1], Kate Revoredo[2], Flávia Maria Santoro[3]

[1] Federal University of the State of Rio de Janeiro, Rio de Janeiro, Brazil
[2] Vienna University of Economics and Business, Vienna, Austria
[3] University of the State of Rio de Janeiro, Rio de Janeiro, Brazil
pedro.mello@uniriotec.br, kate.revoredo@wu.ac.at, flavia@ime.uerj.br

Abstract. Business process monitoring aims at maintaining the reliability of process executions. Nevertheless, the dynamic nature of business processes hinders a proactive scenario in which risk mitigation actions can occur before the facts that put the process at risk. We argue that understanding failures behavior allows proactive actions. Analysing historical data of processes executions supports the identification of situations and patterns of failure behavior. In this paper, we present an experiment in which a combination of well-established techniques from Data Mining and Process Mining fields are applied to an incident management process. The results obtained show that it is possible to identify failures in order to reach for a proactive risk mitigation scenario.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: Process mining, Data mining, Failure prediction, Case study

1. INTRODUCTION

Business processes describe the strategic and operational objectives of an organizational environment through coordinated activities [Dumas et al. 2018]. Business Process Management (BPM) deals with managing the processes of an organization, ensuring consistent outcomes and possibilities of improvement. It is a continuous cycle comprising 6 phases: process identification, discovery, analysis, redesign, implementation, monitoring and controlling. Process monitoring is one of these phases in which data is collected and analyzed providing information on the process performance in order to start actions that might guarantee the achievement of goals. Given the dynamic nature of processes and the lack of knowledge on situations that influence its performance and flow of activities, decision making can be compromised [Hompes et al. 2017]. These aspects increase the complexity of proactive strategies. In a reactive scenario, decisions and risk mitigation approaches are applied only after the occurrence of undesired events. On the other hand, a proactive scenario allows decision making before such events take place. A proactive strategy could be based on an empirical reality estimative, therefore being considered predictive. Moreover, estimations could be extracted from process performance indicators, which are in general defined as metrics that helps identifying performance problems [Dubois and Pohl 2017].

The identification of performance problems can be achieved through process monitoring activities, which is considered as a field of study in the Process Mining research area. This field of study aims to measure the performance of processes without the need to build or use a process model [van der Aalst 2016]. This premise points to the challenge to providing operational support to detection, prediction

and recommendation, which are among the eleven open challenges in the Process Mining [van der Aalst et al. 2012]. Process Mining aims to discover, monitor and improve business processes by extracting knowledge from data; therefore, it comprises techniques and concepts of Business Process Management, Data Mining and Machine Learning [van der Aalst 2016].

Several process mining techniques have been proposed to boost process management, both in scenarios of diagnosis and improvement of business processes [Weske et al. 2018]. Diagnostic scenarios search for explanations of the existence of certain characteristics of the process instances and, in some cases, use the acquired knowledge to estimate an empirical reality and create predictive approaches.

Despite the potential to learn the behavior of any process attribute, predictive approaches in process management are normally related to issues associated to the process cycle, such as predicting the time to complete an instance or which would be the next activity to be performed. Little attention has been given to the prediction of the failures of the process instances. So, aiming to fill this gap, in this paper we propose a method based on process mining techniques to support the prediction of business processes performance indicators.

The performance indicator can determine whether a particular process instance achieves the strategic and operational objectives of the process model. Thus, if the instance does not achieve the expected results, then it represents a risk to the business. In this paper, business risk is determined by the service level agreement (SLA), which is a commitment made by a service provider to a customer [del Río-Ortega et al. 2015]. A failure to comply with the SLA can result in financial losses for the organization. Thus, we argue that whenever there is evidence of non-compliance with the SLA, the particular instance has failed in terms of the objective idealized by the business process.

The incident management in Information Technology (IT) is used as a scenario for this research. The method and results presented are based on the following research question: *How to combine and apply Data Mining and Process Mining techniques to predict failure in the domain of incident management?*

The method proposed consists of techniques for discovering process models, identifying and grouping different instances of a process together with techniques for predicting the failures in the process instances. The evaluation of the method was made with real data from the information technology department of a Brazilian organization. The main contribution of this paper is the definition and evaluation of risk mitigation strategies during the execution of processes in a real scenario.

The paper is organized as follows. Section 2 presents the background knowledge. In Section 3, the works related to ours are described. Section 4 describes the scenario, experimental setting and the prediction mode. Section 5 discusses the results. Section 6 presents conclusions and future perspectives of this research.

2. PRELIMINARIES

This section presents formal definitions for the terms adopted in this paper. It also presents the concept of event summarization and the vector of variables as an important step for the application of non-process-aware techniques. Non-process-aware techniques are applied to data that do not convey a linked structure of events, i.e., it does not consider the transition between the activities of the process and the related directed graphs. On the other hand, process-aware techniques are characterized by considering the structure of the process [Marquez-Chamorro et al. 2017].

2.1 Event Log

The input data for the application of any Process Mining technique is an event log. The log is a sequence of events (or activities) originated from multiple process instances. Events are ordered over

time and have a set of associated attributes [van der Aalst 2016]. The basic structure of the event log consists of three attributes: case identifier (process instance), trace, event identifier. A formal definition for events and attributes is presented in Definition 2.1. For *Cases*, *Trace* and *Event Log* the Definition 2.2 is stated. These definitions are based on [van der Aalst 2011; 2016].

Definition 2.1. (Event, Attribute). Let $\mathcal{E}$ be the set of all events identified and AN be the set of all attributes. For an attribute $x \in AN$, let χ_x be its universe, i.e., the set of all possible values for x. Given a $\mathcal{E}$ and an attribute $x \in AN$. The function $\#x : \mathcal{E} \rightarrow \chi_x \cup \perp$ maps the attribute value x to any event $e \in \mathcal{E}$. The function $\#_x(e) = \perp$ maps all attributes x not defined for e.

Definition 2.2. (Case, Trace, Event Log). Let $\mathscr{C}$ be the set of all process instances identified and AN be the set of all attributes. For any process instance $c \in \mathscr{C}$ and an attribute $x \in AN$. The function $\widehat{\#_x} : \mathscr{C} \rightarrow \chi_x \cup \perp$ maps the value of the attribute x to a process instance c (just like events, process instances have attributes). A *trace* is a mandatory attribute of a process instance and represents a finite sequence of events $t \in \mathcal{E}$ so that there is not two equal events, i.e., for $1 \leq i < j \leq |t|, t(i) \neq t(j)$. Moreover, the events in a *trace* follow an ascent order in time if the timestamp attribute exists, i.e., $1 \leq i < j \leq |t|, \#_{time}(t(i)) \leq \#_{time}(t(j))$.

For a process instance c, $\mathcal{E}_c$ is the set of events that belongs to c, i.e., $\mathcal{E}_c = \widehat{\#}_{trace}(c)$. An *event log* is a set of process instances $\mathcal{L} \subseteq \mathscr{C}$ so that each event occurs only once, i.e., for any two process instances $c_1, c_2 \in \mathscr{C}, c_1 \neq c_2 : \mathcal{E}_{c1} \cap \mathcal{E}_{c1} = \emptyset$.

The event log can reveal the dynamic behavior of the process instances. This dynamic may in turn uncover that the process needs adaptation. This issue is discussed in the next section.

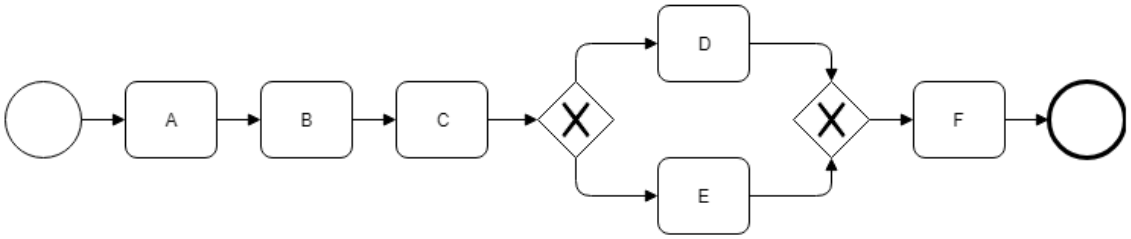
2.2 Process Dynamics

Business processes are dynamic by nature [van der Aalst 2016]. This definition alone already implies that changes can occur during the execution of the processes. Thus, processes may undergo variations to remain operative in certain situations. These variations can manifest themselves in relation to the customer's characteristics, geographical differences, climate changes, in addition to other examples that can be found in the literature [Syamsiyah et al. 2017]. These variations might result in the heterogeneous types of process instances, which are called process variants. According to La Rosa et al. [2013], a process variant is a specific sequence of activities where the path between the first and the last activity forms a single one, i.e., a path without branches.

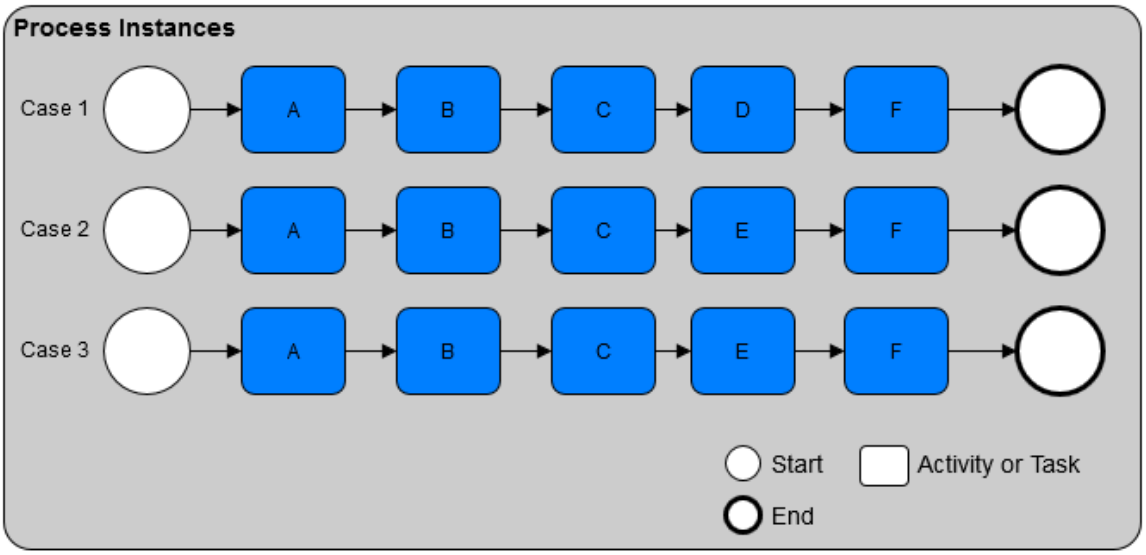
It is important to guarantee this heterogeneity because, through the process variants, it is possible to identify which patterns, in terms of operational flow, can deliver good results. La Rosa et al. [2013] state that the frequency of the process variants allows to identify patterns capable of distinguishing outliers i.e. patterns, instances or samples that differ considerably from all the others; exceptions i.e. exceptional situations aimed at meeting specific needs, incomplete or running instances. They also allow to identify points of customization of a process. The flexibility of a process can also be measured and evidenced by analyzing the process variants.

Figure 1 depicts an example of a process model (see Figure 1a) and a set of process instances (see Figure 1b). We can observe that some instances have events that are not present in others, e.g. event D appears in Case 1 while event E in Cases 2 and Case 3. This is due to the branching of the process. Case 1 has one variant and Case 2 and Case 3 have a second variant.

From the definitions and examples presented, we argue that identifying process variants helps to determine which ones have good results, i.e. the ones that generally achieve the intended goals. Nevertheless, this heterogeneity can increase the complexity of process management by analysts and managers [Hompes et al. 2017]. The dynamics to which business processes are subject is not related



(a)



(b)

Fig. 1: Process model and its instances. (a) Process model idealized through its operational flow, (b) Instances of the process.

to their flows only, but also to a whole operational context. Changes can be caused by the entry of new customers, resources and even operational errors.

2.3 Event Encode and Feature Vector

The summarization aims to obtain a representation of the event history as a vector of variables, also known as Event Encode [Marquez-Chamorro et al. 2017]. There are different techniques that can be applied in the summarizing of process events, or more specifically in Event Encoding, such as, (*i*) *boolean encoding* (Table Ia); (*ii*) *frequency-based encoding* Ib) and (*iii*) *index-based encoding* (Table Ic). Table I presents examples of these representations defined in [Leontjeva et al. 2015].

Considering Table I, the forms of encoding differ. For *boolean encoding* and *frequency-based encoding*, there is a sequence of events described in a feature vector in which each variable corresponds to a class of events (or activities) registered in the log, i.e., each variable is defined as $e_1, e_2, \ldots, e_n \in \mathcal{E}$, where the values assigned to the variables of the encoding are represented in general as:

Table I: Examples of Event Encode (adapted from [Leontjeva et al. 2015])

(a) *boolean encoding*

case	maintenance	installation	email	⋯	label
c_1	1	0	0	⋯	false
c_2	0	0	1	⋯	true
⋮					
c_k	0	1	0	⋯	false

(b) *frequency-based encoding*

case	maintenance	installation	email	⋯	label
c_1	1	0	0	⋯	false
c_2	0	0	3	⋯	true
⋮					
c_k	0	1	1	⋯	false

(c) *index-based encoding*

case	$event_1$	⋯	$event_m$	$resource_1$	⋯	$resource_m$	label
c_1	maintenance	⋯	email	Anna	⋯	Anna	false
c_2	maintenance	⋯	installation	Anna	⋯	John	false
⋮							
c_k	maintenance	⋯	email	Anna	⋯	Anna	false

Let $e_1, e_2, \ldots, e_n \in \mathcal{E}$ for a *case* c, with:

$$c_i = \text{a process instance with index i}$$
$$v_i = \text{feature vector}$$

So that,

$$v_{ij} = \begin{cases} 1, & \text{if } e \text{ is present in } c_i \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Where,

$$v_i = [v_{i1}, v_{i2}, \ldots, v_{ij}]$$
$$v_{ij} = \text{ corresponds to a class of events } e \in \mathcal{E}$$

As a result, each event or activity is represented by means of a pertinence relation in the feature vector. For the *frequency-based encoding*, the flow is represented by the frequency of each event and not by means of a simple relation of pertinence with only two possible values (1 or 0). For the *index-based encoding*, data associated with events is divided into static and dynamic: static data is the same for all events and dynamic data is different for each event.

3. RELATED WORK

The related works refer to the central topic of this paper, i.e., business processes prediction. In this context, some different approaches were identified in the literature: (i) prediction of events or activities; (ii) prediction of process models; and (iii) prediction of business process performance indicators. For each category, we observe the use of two possible techniques: (i) process-aware and

(ii) non-process-aware. Process-aware techniques of are characterized by considering the structure of the process [Marquez-Chamorro et al. 2017], which means that the transition between activities and the respective directed graphs are considered. On the other hand, non-process-aware techniques are applied to data that do not refer to a chained structure of events. Even though they are different approaches, we highlight works that have achieved good results with both of them. [Jarke et al. 2014], [Leontjeva et al. 2015] and [Márquez-Chamorro et al. 2017] are non-process aware techniques and [Breuker et al. 2016] is a process aware technique.

In [Jarke et al. 2014], the authors emphasize that the main problem related to process monitoring tasks is the fact that reactive approaches only identify operational failures in the process after their occurrence. As a solution, the authors suggest a predictive approach consisting of using a *Decision Tree* algorithm to predict violations of business restrictions. It was evaluated through a case study in a medical diagnosis process domain. The evaluation used the most common metrics within the machine learning area such as *Accuracy*, *Precision*, *Recall*, *Specificity* and the form of segmentation and separation between training and test data set follows the ratio of 80% - 20% respectively. The authors also use 10 - *fold cross-validation*. Their scientific contribution is a *framework* capable of estimating the process compliance probability.

Leontjeva et al. [2015] highlight the difficulty in predicting the possible results and expected outputs of a running process (an incomplete instance). The authors propose different forms of event encoding in order to enable the use of well-established algorithms of Machine Learning. The proposal uses *index-based encoding* enriched with probabilistic information from the events. They also performed a case study with two real processes: the first process consists of the historical patient record of a hospital and the second of an insurance company. The authors use the area under the ROC curve defined from the confusion matrix as a method for evaluating the proposal. The main contribution was to show that enriching the forms of encoding with probabilistic information improves in some cases the reliability of the prediction.

Márquez-Chamorro et al. [2017] also agree about the complexity of managing processes proactively. They proposed to discover a set of patterns by adopting an Event Window strategy with a genetic algorithm applied to two incident management processes log. They consider a classification of the SLA for a predictive scenario in order to improve or provide operational support in risk mitigation strategies. The authors achieved best results when compared to some other techniques already mentioned in the literature.

According to Breuker et al. [2016], predictive monitoring techniques need to be improved. Once this prediction is possible, an information system can issue alerts about undesirable events that may occur in the future. Their main goal was is to develop a process modeling technique in a predictive manner based on techniques Process Mining and Grammatical Inference [Wieczorek 2017]. They built an artifact to make the prediction of process models which was evaluated the measures *Accuracy*, *Precision*, *Recall* and *Specificity*.

Recent works are focused on specific problems such as next activity, remaining time, and failure. Mehdiyev et al. [2020] addressed the prediction of the next process event as a classification problem. The authors propose a multi-stage deep learning approach following a feature pre-processing stage with n-grams and feature hashing with the learning model consisting of an unsupervised pre-training component with stacked autoencoders and a supervised fine-tuning component. Borkowski et al. [2019] explained how to employ event-based failure prediction in business processes by employing machine learning techniques with various types of events. The results showed that this approach was able to detect errors and predict failures with high accuracy. Verenich et al. [2019] proposed a taxonomy and a cross-benchmark comparison of methods for remaining time prediction of business processes based on a systematic literature review.

The techniques presented were used in a complementary way in this paper. The encode techniques,

as explained in Section 2.3 and according to the work of Leontjeva et al. [2015], were used in the activity that precedes the application of the training algorithms. The way to calculate the indicators as attributes of the events as well as attributes of the processes instances is present in the work of Márquez-Chamorro et al. [2017]. The evaluation measures for this work follow the same measures for the work presented by Breuker et al. [2016].

4. EXPERIMENT DESIGN

The main objective of this paper is to discuss a method for the prediction of failures in business processes in real case. Process failures are defined through the SLA already established by the organizational environment.

This section presents the proposed steps for modeling the prediction of failure situations. These steps involve the combination of different techniques from the Data Mining and Process Mining areas. They also involve pre-processing techniques that aim to obtain a final dataset derived from initial raw data to obtain new representations that better describe an empirical reality and serve as input to the algorithms chosen. The proposed method consolidates the steps mentioned above and is shown in Figure 2.

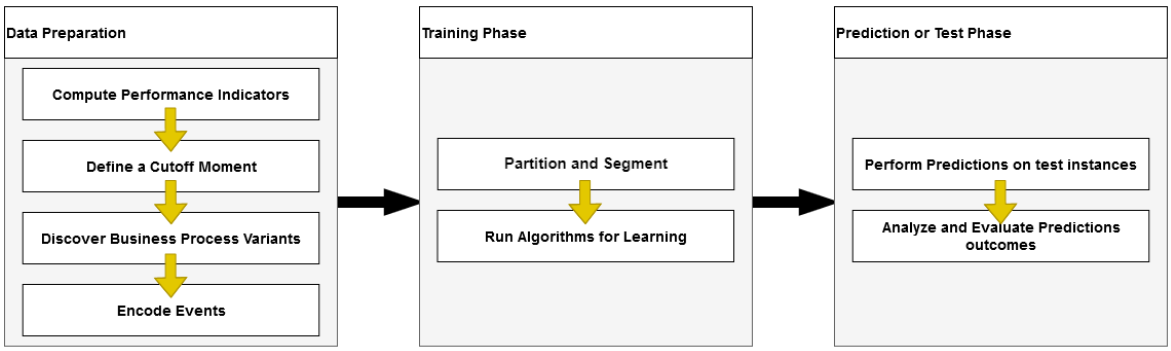


Fig. 2: Phases for predicting failure in business processes.

The following sections present the application of the method in the real setting scenario.

4.1 Scenario Description

This case considers an event log that depicts the incidents recorded by the information technology department of a Brazilian company in the years 2015 and 2016. Incidents are defined by any operational failure of an IT service, i.e. any failure involving *software*, *hardware* and any other peripherals such as servers, printers and telephony. In this scenario, the existence of SLA with a problem resolution time set greater than 720 minutes is considered an instance with a failure situation. A problem resolution time is the time elapsed from the moment a ticket for solving the incident is opened until it is closed. By making the failures explicit in the log, it would be possible to measure the differences between the context in which the process was conceived against the current state of the business and the service offering.

The scenario considered has a total of 294,628 events and 7,259 process instances. The Event log has the following attributes:

Attributes of the process:

—**Case**. identifies the incident.

—**OpenTime**. time the incident was open.

—**CloseTime**. time the incident was concluded.

—**Customer**. identifies the client or responsible for opening the incident in the system.

—**Service**. indicates if the incident is related to update of software, hardware or local network.

Attributes of process events:

—**EventId**. identifies the event in a transactional level.

—**EventName**. characterizes the event of activity of the process.

—**Article**. Indicates if there were messages exchange among the participants, i.e., between who open the incident and the person responsible for solving the incident.

—**Priority**. indicates the level of priority of the incident.

—**TimeStamp**. date and time the event was registered in the system; allows a topological order of the events of a process instance.

4.2 Data Preparation

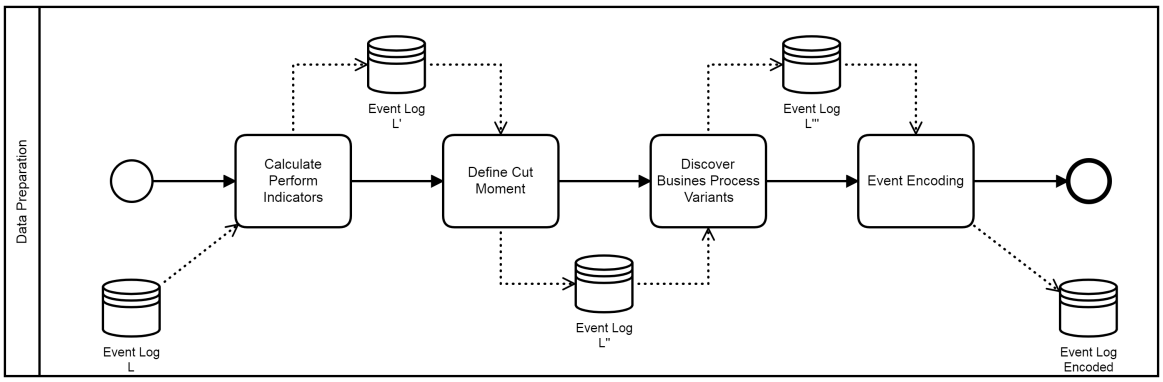


Fig. 3: Data preparation for the case study.

Figure 3 presents the Data Preparation Phase of the proposed method applied to our scenario. As it can be seen, this phase consists of four tasks: (*i*) *Calculate performance indicators* for each instance of the process; (*ii*) *Define a Cutoff moment* for predictive modeling; (*iii*) *Discover Business Process Variants* and enrich the event log with these information (*iv*) *Encode Events* obtain a coded representation of the event log.

In the first task, the indicator is obtained through two steps: (*i*) calculate the time that elapsed between the first event of the process instance and the iteration event of the moment; and (*ii*) test if elapsed time is greater than 720 minutes. After (ii), an attribute is assigned to each process instance indicating that the time limit has been exceeded. We call this attribute a timeout overrun attribute by a performance indicator that presents a situation of failure or, in a more simplified way, an instance failure indicator.

Figure 4 shows the cutoff moment, which is determined by means of the event preceding the first event with the failure identifier attribute (represented in red). After identifying the cutoff moment, all events that precede it are removed, as the next events are no longer relevant to the prediction given that the failure situation has already occurred. Then, all instances have the cutoff moment parameterized according to the minimum number of events of the instances with failure indicator.

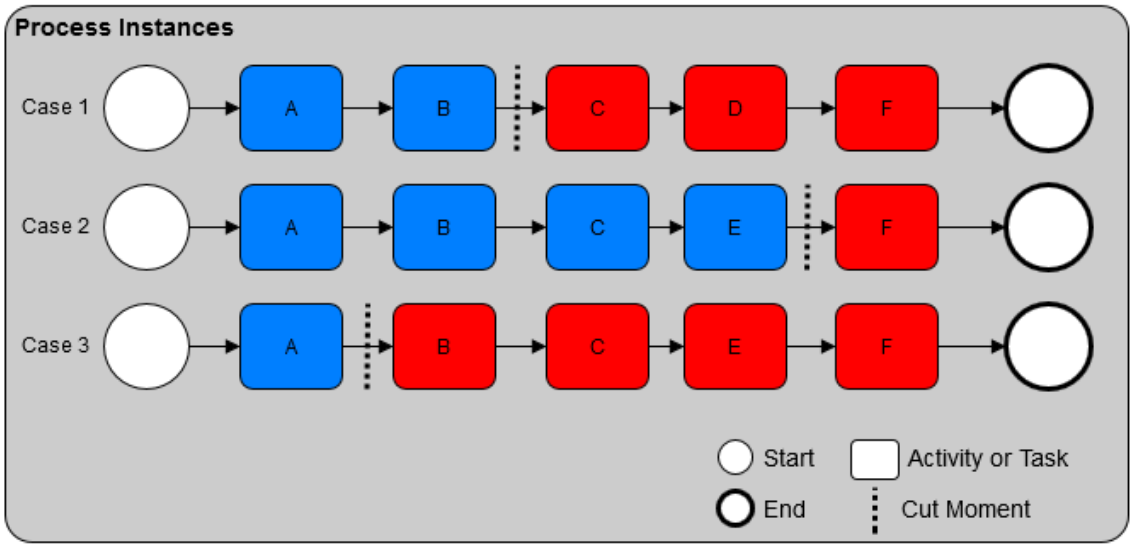


Fig. 4: Cut applied to the process instances that fail in different activities or events.

In this way, we can balance the number of events for all instances, regardless of the indicator. Thus, all instances are treated as incomplete, i.e., as if they were still running.

Once this is done, the *Business Process Variant Discovery* task identifies which variants potentially achieve their goals. This task uses the event log obtained after the cutoff ($\mathcal{L}''$). This experiment used the *Fuzzy Miner* [Günther and van der Aalst 2007] algorithm. Fuzzy Miner was proposed to overcome the problem of discovering "spaghetti-like" models. It is an approach configurable and allows for different faithfully simplified views of a particular process. Besides, it has been proved to provide meaningful abstractions of operational processes in diverse domains [Günther and van der Aalst 2007]. Fuzzy miner [Günther and van der Aalst 2007] was the first mining algorithm to introduce the "map metaphor" to process mining. The fuzzy miner is a process discovery algorithm that discovers process graphs (and not Petri nets, as the traditional ones). Process graphs are especially useful to explore and get initial insights in the dataset. Most process mining techniques follow an interpretive approach, i.e., they try to map the behavior found in the log to process design patterns (for example, if a split node has AND or XOR semantics). The Fuzzy approach focuses on mapping the behavior found in the high-level log. Thus, creating a preliminary process model (not simplified) is simple: all classes of events found in the log are converted into nodes (activities), whose relevance is expressed by unary significance. For each observed precedence relation between event classes, a corresponding directed edge is included in the process model. This edge is described by its binary significance and by the correlation of the ordering relationship it represents. Subsequently, three transformation methods are applied to the process model, which will successively simplify specific aspects of it. The first two phases, conflict resolution and edge filtering, remove edges (that is, precedence relations) between nodes (activities), while the final phase of aggregation and abstraction removes and / or groups less significant nodes. Removing the edges of the model first is important due to the less structured nature of real-life processes and the measurement of long-term relationships. The initial model contains sorting relationships that may not correspond to valid behavior and need to be discarded (Günther and van der Aalst, 2007). Mining Fuzzy's flexible approach adaptively simplifies the mining process models.

At the end, there is a log $\mathcal{L}'''$ with the following attributes added:

—**Events**. number of events of the process instance.

—**Variant**. process variant identifier.
—**Started**. starting time of the process instance event.
—**Finished**. finishing time of the process instance event.
—**Duration**. duration of a process instance.

Finally, the task *Event Encoding* aims to obtain a representation of the event log as a vector. Different techniques can be applied to *Event Encoding*, for example, (*i*) *boolean encoding*; (*ii*) *frequency-based encoding* and (*iii*) *index-based encoding*, presented in Section 2.3.

To obtain the final representation of the log $\mathcal{L}'''$ as a vector of variables, it is necessary to treat the attributes that vary between events of the same instance. Therefore, the proposed method combines two techniques: *boolean encoding*, where a pertinence relation is assigned to each variable that corresponds to a class of events (or activities) in the process; and *frequency encoding*, where each class of events that established the pertinence relations is added. As a result, event information is not lost and through *frequency encoding* it is possible to identify whether there are changes to the same attribute at different times of a process instance.

4.3 Training

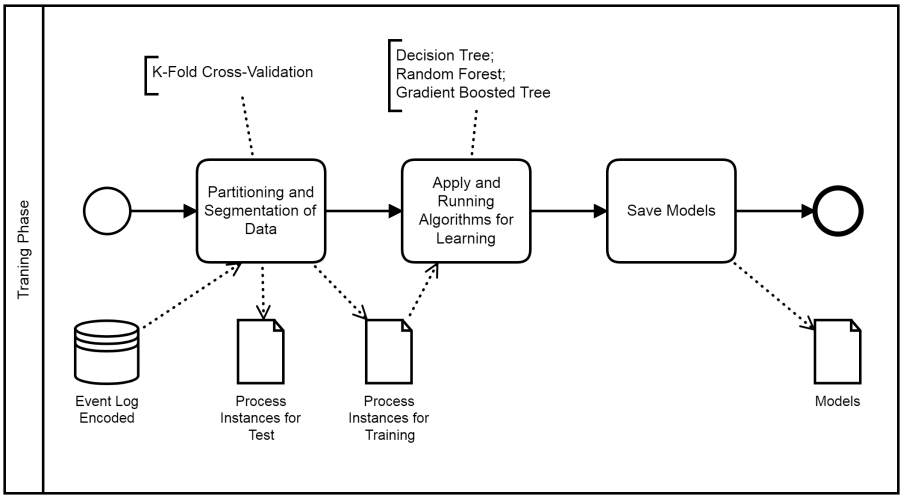


Fig. 5: Learning phase steps.

In the Training phase, a prediction model is learned based on the encoded event log derived from the pre-processing (data preparation) phase. Figure 5 depicts the details of this phase.

Considering Figure 5, the flow of tasks in the training phase is given by:

—**Perform data segmentation operations.** In this task, segmentation follows a strategy of separating the data set between training instances and test instances. The highlighted strategy is *K-Fold Cross-Validation* but other techniques like *Holdout Cross-Validation* and *Leave-one-out Cross-Validation* are also applicable, more information on these techniques can be found at [Kohavi 1995] .;
—**Apply learning algorithms.** In this task, machine learning algorithms are applied. These algorithms have as input the set of training instances, obtained through previously. Some of the most used algorithms are *Decision Tree*, *Random Forest* and *Gradient Boosting Tree*. These algorithms were part of the scientific experimentation in this paper.;

—**Save model.** In this task, the trained model is saved for future evaluations in order to support the choice of the best trained model. Depending on the segmentation strategy adopted, several models are generated.

First, the data should be split in training and testing data. A *10-Fold Cross-Validation* [Kohavi 1995] was applied. For the learning step, machine learning algorithms for learning *Decision Tree*, *Random Forest* and *Gradient Boosting Tree* models were considered. The learning models were saved for future evaluations in order to support the choice of the best trained model. Depending on the segmentation strategy adopted, several models are generated.

4.4 Prediction

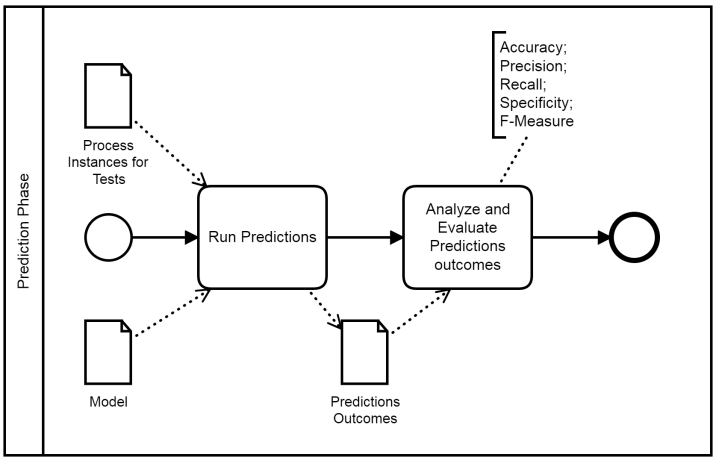


Fig. 6: Prediction, evaluation and test steps.

The prediction phase, also used as evaluation and testing, deals with the flow of activities that predicts the failure attribute of process instances that did not belong to the set of instances used in the training phase.

Figure 6 depicts how the prediction was performed. Predictions are made on the test data and different evaluation measures were used to indicate which could be considered the best model for our scenario. Considering Figure 6, the flow of activities in the prediction, evaluation and test phase is given by:

—**Run predictions.** In this task, predictions are made on the test data set obtained through the segmentation strategy adopted producing the results, referring to the variable that indicates failure, which the model estimates to be as assertive;

—**Evaluate results.** In this task, an assessment is performed by comparing the results of the model estimated with the actual results of the test instances. Some strategies that could be adopted are: *Accuracy*, *Precision*, *Recall*, *Specificity* and *F-Measure*.

Some computational tools can be used support fulfilling the tasks of model construction. For example, tools like *WEKA* [1], *RapidMiner* [2] and *Orange* [3] are some examples of tools that implement

[1]More information can be found at `https://www.cs.waikato.ac.nz/ml/weka/`
[2]More information can be seen at `https://rapidminer.com/`
[3]More information can be seen at `https://orange.biolab.si/`

machine learning algorithms as much as the algorithms that allow you to segment data. It is also possible to use programming languages in conjunction with libraries that can be imported into the source code in order to follow the *pipeline* suggested using programming such as *scikit-learn* [4] or *SparkMLlib* [5].

4.5 Evaluation

The evaluation of the results is conducted through a quantitative analysis. According to Recker [2012], the quantitative method describes a set of techniques that aims to answer research questions under a numerical emphasis. The choice of the quantitative method by means of experiments is due to the fact that the research is based mainly on data and is intended to examine the effectiveness of the prediction of the experiments.

To evaluate the confidence of the results obtained, the experimental research adopts the following measures: Accuracy [Witten et al. 2016], Precision [Fawcett 2006], Recall [Fawcett 2006], Specificity [Fawcett 2006] and F-Measure [Witten et al. 2016].

These measures are widely used to evaluate the results of machine learning techniques application in the literature [Marquez-Chamorro et al. 2017], and they are balanced in comparative terms between true positive rates and false positive rates. This balance is also found for true negative and false negative rates. That said, the prediction method adopted by this work consists of the classification of a Boolean (binary) variable that denotes a situation of failure or success and, therefore, the choice for these measures is natural given the context presented.

—*Accuracy*. The number of instances correctly classified in proportion to the total number of instances;
—*Precision*. The ratio between the number of true instances correctly predicted and the total of true instances correctly predicted and false instances incorrectly predicted;
—*Recall*. The ratio between the number of true instances correctly predicted and the total of true instances correctly predicted and incorrectly predicted;
—*Specificity*. The ratio of the number of correctly predicted false instances to the total number of correctly predicted and incorrectly predicted false instances;
—*F Measure*. The weighted average between the precision and the recall of each category, in the case of the domain, these are failure situations.

The use of a confusion matrix also allows us to observe the model's error rate intuitively. The main idea of a confusion matrix is to compare the predictive values with the real values of the model. That said, with the use of the confusion matrix it is possible to observe whether the difference between the expected values and the actual values exceeds acceptable limits.

5. RESULTS AND DISCUSSION

The application of the Fuzzy Miner algorithm resulted in the discovery of a process model with many branches. It shows that the process is able to adapt to diverse situations. Since, by definition, not all operational flows have good results, process variables are identified by grouping the operational flow of the process instances. In this way, it is possible to identify which flows (which paths without branches) present good results.

Figure 7 depicts a scatter plot allowing the analysis of the variants. It is possible to see if Some variants of the process have failed more often and, thus, to conclude which ones may be more

[4]More information can be seen at `https://scikit-learn.org/stable/`
[5]More information can be seen at `https://spark.apache.org/docs/latest/ml-guide.html`

susceptible to failure. Three variants stand out (1, 3 and 5) in terms of volume of process instances where the 1 and 3 variants have an average elapsed time of less than 720 minutes, that is, less than 12 hours . On the other hand, the process variant identified by the number 5 is characterized by having a considerable volume of process instances (122 instances) that have a time greater than 377 hours.

The process variants that appear only once throughout the observed period were disregarded because they did not present instances in both events in the event log (success or failure).

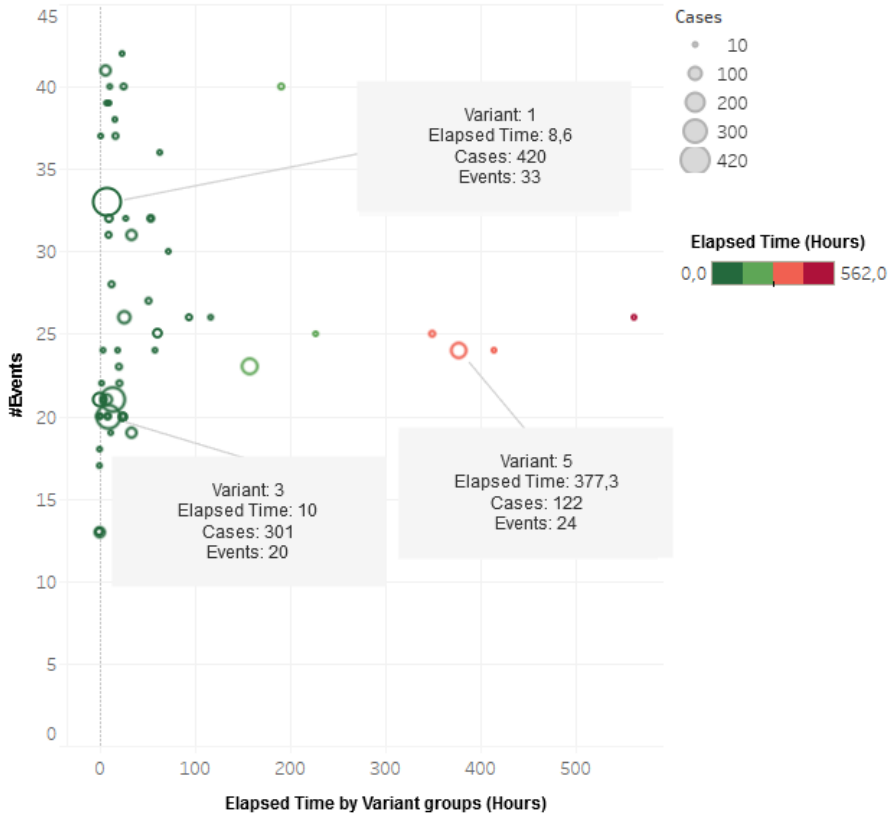


Fig. 7: Scatter plot used to verify the ability to discriminate the data set through process variants. On the x-axis: total events that a process variant has; y-axis: Average elapsed time.

The analysis of the process variants presented in Figure 6 precedes the cutting off task. The cutting off task resulted in process variants whose minimum size is 8 events. This implies that after eighth event many instances failed. Thus, all instances of the experiment were limited to a total of 8 events from the opening of the incident. After the cutting step, the experiment follows to the Event Encoding taks. The form of Event Encoding considered was the combination of two techniques described in Section 2.3. The first is to apply boolean encoding for dynamic attributes and then, for each event of a process instance, apply the frequency encoding to the results of the boolean encoding. Thus, static information and dynamic information are not lost, and through frequency encoding, it is possible identify if there are changes of the same attribute at different times of a process instance, being a way to deal with dynamic attributes.

Then, the learning algorithms were applied to the Event Encoded and the results are presented in Tables 2 and 3. To assess the confidence of the results obtained after applying the algorithms, we adopted the following measures: Accuracy, Precision, Recall, Specificity and F-Measure [Fawcett

2006; Witten et al. 2016]. Table II depicts the average for 10-fold *cross-validation* with the ratio of 90% – 10% for training and test data respectively.

Table II: Summary of the results from failure prediction.

	accuracy	*precision*	*recall*	*specificity*	*Fmeasure*
DT	65.17%	61.93%	79.52%	50.69%	59.16%
RF	67.07%	68.08%	64.84%	69.32%	67.70%
GBT	72.22%	71.44%	74.44%	69.98%	71.49%
mean	68.15%	67.15%	72.93%	63.33%	66.11%
stdev	3.64%	4.82%	7.45%	10.95%	6.31%

Table III: Confusion matrix related to the results of: (a) DT; (b) RF; (c) GBT.

(a)

		Pre-known		
		Positive	Negative	Total
Prediction	Positive	986	606	61.93%
	Negative	254	623	71.04%
	Total	79.52%	50.69%	

(b)

		Pre-known		
		Positive	Negative	Total
Prediction	Positive	804	377	68.08%
	Negative	436	852	66.15%
	Total	64.84%	69.32%	

(c)

		Pre-known		
		Positive	Negative	Total
Prediction	Positive	923	369	71.44%
	Negative	317	860	73.07%
	Total	74.44%	69.98%	

Considering Table II, the GBT algorithm stands out in three results when compared to DT and RF. These results are: (*i*) *accuracy*, (*ii*) *precision* and (*iii*) *Fmeasure*. The RF algorithm stands out in three results when compared to DT. We attribute the superiority of GBT and RF compared to DT to the technique known as Methods of *Ensemble*.

The *Ensemble* methods emerge from the hypothesis that the combination of different base algorithms can improve the accuracy of a given problem [Hastie et al. 2009]. This hypothesis is valid for the domain explored by this article. Still regarding the results presented in Table II, we attribute the superiority of GBT if compared to RF to the technique known as *Boosting*. *Boosting* techniques are

found within the *Ensemble* methods and consist of using weighted versions of the same training set for the base algorithms [Schapire and Freund 2012; Hastie et al. 2009].

Table III shows the confusion matrix obtained for each of the experimentation algorithms and their respective rates. The matrix is organized with the pre-known data obtained by calculating the failure indicator and the values resulting from the prediction.

Table IV shows the setup parameters of each algorithm.

Table IV: Input parameters for each algorithm we used.

(a)

Decision Tree	
criterion	gini index
maximal depth	10
confidence	0.25
minimal gain	0.01
minimal leaf size	2
minimal size for split	4
number of prepruning	3

(b)

Random Forest	
criterion	gini index
number of trees	100
maximal depth	10
min rows	1.0
number of bins	20
learning rate	0.1
sample rate	1.0
minimal leaf size	2
number of prepruning	3

(c)

Gradient Boosted Tree	
number of trees	100
maximal depth	10
min rows	1.0
number of bins	20
learning rate	0.1
sample rate	1.0
minimal leaf size	2
number of prepruning	3

The limitations of this research are related to two issues. First, only one scenario was used to test the proposal. So, it is not possible to generalize the results, besides, it could introduce some bias to the results and conclusions. We argue that at least in the IT problem solution process, the scenario is very typical and as so, it represents quite well the domain. Second, we used the attributes made available by the company, but other attributes, such as, the specific characteristics of the customers and equipment, could improve even more the results.

6. CONCLUSION

In this paper, we present an empirical study that applies different techniques in the area of Data Mining and Process Mining in a complementary way. The combination of these techniques is proposed in order to fill some gaps in the steps for the prediction a variable or process attribute. We refer to this variable as a process performance indicator which indicates whether the process instance has achieved the expected performance in terms the strategic and operational objectives. Thus, we contribute to improve the transition between a reactive scenario to a predictive scenario.

The Event Encoding proposed follows a special approach, where the activities are summarized using the results of the *Fuzzy Miner* to avoid the need to apply the *index-based encoding* technique, since, the flow is represented directly in the process variant information. The techniques *boolean encoding* and *frequency-based encoding* are used for the dynamic attributes.

Regarding the prediction of the failures of the process instances, we identify the potential of three algorithms to address with the lack of internationally accepted *benchmarks*. The technique presented by this paper is non-process-aware and the main difference in relation to the works presented in Section 3 is in the use of data from process variants.

Finally, we also highlight the operational flows applied to the case study (Section 4.3) which proved to be very flexible for the application of different techniques and algorithms. We concluded that the work went through several challenges still open in the Process Mining area to reach the prediction, such as the challenges listed by 1, 2, 3, 4, 6, 7, 8 and 9 in the literature reference given by [van der Aalst et al. 2012].

As a future work, we intend to implement the method shown here as a plug-in for *ProM Software*[6] and apply the method presented in other domains and obtain new representative *benchmarks* also considering the use of process aware techniques.

REFERENCES

Borkowski, M., Fdhila, W., Nardelli, M., Rinderle-Ma, S., and Schulte, S. Event-based failure prediction in distributed business processes. *Information Systems* vol. 81, pp. 220–235, 2019.

Breuker, D., Matzner, M., Delfmann, P., and Becker, J. Comprehensible predictive models for business processes. *MIS Quarterly* 40 (4): 1009–1034, 2016.

del Río-Ortega, A., Gutiérrez, A. M., Durán, A., Resinas, M., and Ruiz-Cortés, A. Modelling service level agreements for business process outsourcing services. In *Advanced Information Systems Engineering*, J. Zdravkovic, M. Kirikova, and P. Johannesson (Eds.). Springer International Publishing, Cham, pp. 485–500, 2015.

Dubois, E. and Pohl, K., editors. *Enriching Decision Making with Data-Based Thresholds of Process-Related KPIs*. Springer International Publishing, Cham, 2017.

Dumas, M., Rosa, M. L., Mendling, J., and Reijers, H. A. *Fundamentals of Business Process Management*. Springer Publishing Company, Incorporated, 2018.

Fawcett, T. An introduction to roc analysis. *Pattern recognition letters* 27 (8): 861–874, 2006.

Günther, C. and van der Aalst, W. Fuzzy mining–adaptive process simplification based on multi-perspective metrics. *Business Process Management*, 2007.

Hastie, T., Tibshirani, R., and Friedman, J. The Elements of Statistical Learning. *Bayesian Forecasting and Dynamic Models* vol. 1, pp. 1–694, 2009.

Hompes, B. F. A., Maaradji, A., La Rosa, M., Dumas, M., Buijs, J. C. A. M., and van der Aalst, W. M. P. Discovering Causal Factors Explaining Business Process Performance Variation. In *CAiSE*. Vol. 7328. pp. 177–192, 2017.

Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., and Horkoff, J., editors. *Predictive Monitoring of Business Processes*. Springer International Publishing, Cham, 2014.

Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, 1995.

La Rosa, M., van der Aalst, W. M. P., Dumas, M., and Milani, F. P. Business process variability modeling: A survey. 50 (1): 1–45, 2013.

Leontjeva, A., Conforti, R., Di Francescomarino, C., Dumas, M., and Maggi, F. M. Complex symbolic sequence encodings for predictive monitoring of business processes. In *International Conference on Business Process Management*. Springer, pp. 297–313, 2015.

Marquez-Chamorro, A. E., Resinas, M., and Ruiz-Cortes, A. Predictive monitoring of business processes: a survey, 2017.

Márquez-Chamorro, A. E., Resinas, M., Ruiz-Cortés, A., and Toro, M. Run-time prediction of business process indicators using evolutionary decision rules. *Expert Systems with Applications* vol. 87, pp. 1–14, 2017.

Mehdiyev, N., Evermann, J., and Fettke, P. A novel business process prediction model using a deep learning method. *Business & information systems engineering* 62 (2): 143–157, 2020.

Recker, J. *Scientific Research in Information Systems: A Beginner's Guide*. Springer Publishing Company, Incorporated, 2012.

Schapire, R. E. and Freund, Y. *Boosting: Foundations and algorithms*. MIT press, 2012.

Syamsiyah, A., Bolt, A., Cheng, L., Hompes, B. F. A., Jagadeesh Chandra Bose, R. P., van Dongen, B. F., and van der Aalst, W. M. P. Business process comparison: A methodology and case study. In *Lecture Notes in Business Information Processing*, W. Abramowicz (Ed.). Vol. 288. pp. 253–267, 2017.

van der Aalst, W. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag Berlin Heidelberg, 2011.

van der Aalst, W. et al. Process mining manifesto. In *Lecture Notes in Business Information Processing*. Vol. 99 LNBIP. pp. 169–194, 2012.

van der Aalst, W. M. *Process Mining: Data Science in Action*. Springer-Verlag Berlin Heidelberg, 2016.

Verenich, I., Dumas, M., Rosa, M. L., Maggi, F. M., and Teinemaa, I. Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10 (4): 1–34, 2019.

Weske, M., Marco Montali, Ingo Weber, and Jan vom Brocke. Business Process Management. In *Business Process Management*, M. Weske, M. Montali, I. Weber, and J. vom Brocke (Eds.). Lecture Notes in Computer Science, vol. 11080. Springer International Publishing, 2018.

Wieczorek, W. Grammatical inference: Algorithms, routines and applications. sci, vol. 673, 2017.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[6] `http://www.promtools.org`.