Enhancing Completion Time Prediction Through Attribute Selection

Claudio A. L. Amaral[1], Marcelo Fantinato[1(✉)], Hajo A. Reijers[2], and Sarajane M. Peres[1]

[1] School of Arts, Sciences and Humanities, University of São Paulo, 1000 Arlindo Béttio St., Ermelino Matarazzo, São Paulo, SP 03828-000, Brazil
{claudio.amaral,m.fantinato,sarajane}@usp.br
[2] Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands
h.a.reijers@uu.nl
http://www.each.usp.br/fantinato, http://www.reijers.com, http://www.each.usp.br/sarajane

Abstract. Approaches have been proposed in process mining to predict the completion time of process instances. However, the accuracy levels of the prediction models depend on how useful the log attributes used to build such models are. A canonical subset of attributes can also offer a better understanding of the underlying process. We describe the application of two automatic attribute selection methods to build prediction models for completion time. The filter was used with ranking whereas the wrapper was used with hill-climbing and best-first techniques. Annotated transition systems were used as the prediction model. Compared to decision-making by human experts, only the automatic attribute selectors using wrappers performed better. The filter-based attribute selector presented the lowest performance on generalization capacity. The semantic reasonability of the selected attributes in each case was analyzed in a real-world incident management process.

Keywords: Process mining · Attribute selection · Incident management · ITIL · Annotated transition systems

1 Introduction

Estimates for the completion time of business process instances are still precarious as they are usually calculated based on superficial and naïve abstractions of the process of interest [1]. Many organizations have been using Process-Aware Information Systems (PAIS), which record events about the activities carried out in the process involved, generating a large amount of data. Process mining can exploit these event logs to infer a more realistic process model [2], which can be used as a completion time predictor [3]. In fact, general data mining techniques and the similar have been applied for different purposes to improve the performance of organizations by making them intelligent [4–6].

E. Ziemba (Ed.): AITM 2018/ISM 2018, LNBIP 346, pp. 3–23, 2019.
https://doi.org/10.1007/978-3-030-15154-6_1

However, specifically in terms of distinct strategies addressing prediction of completion time for business processes, a common gap of is the lack of concern in choosing the input log configuration. It is not common to seek the best subset of descriptive attributes of the log to support constructing a more effective predictor, as happens in [3,7–10]. For an incident management process, for example, some descriptive attributes for each instance process (i.e., for each incident) can be status, severity, symptom, category, impact, assignment group etc.

Two inputs are expected when building a process model as a completion time predictor – an event log and a set of descriptive attributes. Depending on the organizational settings, the number of existing descriptive attributes can be so large and complex that may be unfeasible to use all the attributes. In addition, studies have shown that the predictive accuracy of process models depends on which attributes have been chosen to create them [11]. Therefore, when building a prediction model, one needs to consider that not all attributes are necessarily useful. In fact, according to Kohavi and John [12], a predictor can degrade in performance (accuracy) when faced with many unnecessary features to predict the desired output. Thus, an ideal minimum subset of descriptive attributes should be selected that contains as much relevant information as necessary to build an accurate prediction model, i.e., a canonical subset of descriptive attributes should be selected.

However, a manual selection of a subset of descriptive attributes may be impracticable. In this sense, this paper details a proposal of how to apply two automatic attribute selection methods as the basis for building prediction models[1]. Consider here an event log e composed of a set of categorical descriptive attributes $\mathit{\Delta} = \{a_1, a_2, \cdots, a_m\}$ that characterize the events of a process instance. Consider $\mathit{\Omega}$ a set whose elements are all combinations of attributes in $\mathit{\Delta}$; each combination of attributes $\omega_i \in \mathit{\Omega}$ can be used to generate a model $\theta_i \in \mathit{\Theta}$, where $\mathit{\Theta}$ is a set of models that represent a process under distinct aspects. Consider the process models $\theta_i \in \mathit{\Theta}$ as predictors of completion time, generated on samples e'_i of the event log e; each model $\theta(\omega, e')$ has a particular prediction performance. Consider the prediction error as the measure of performance. The problem of interest in this paper is formulated as

$$\underset{\omega \in \mathit{\Omega}}{\operatorname{argmin}}\, \epsilon(\theta(\omega, e')),$$

where the minimization process looks for a $\omega \in \mathit{\Omega}$ such that $\epsilon(\theta_i(\omega_i, e'_i)) \leq \epsilon(\theta_j(\omega_j, e'_j))\ \forall\, j$, where $i, j = \{1, \cdots, \#\mathit{\Omega}\}$, $i \neq j$ and $\#\cdot$ represents the number of elements in a set.

In this paper, the minimization process is implemented through a filter technique [14] and two wrapper techniques [12] as the attribute selection methods, using heuristic search techniques – a filter with ranking and the wrapper with hill-climbing and with best-first. These classical attribute selection methods are used to automatically determine a canonical subset of descriptive attributes to

[1] This paper details the approach and results published in a summarized preliminary version [13].

be subsequently supplied to the prediction model. Annotated Transition Systems (ATS) [3] were chosen as the prediction model to compare the different techniques used. ATSs are a good example of a prediction model in this context as they largely depend on the attributes used. For the experiments and analyzes reported herein, ϵ is the mean error on time prediction (in seconds), θ is implemented using ATS and e' are samples of an event log from a real-world incident management process.

The approach discussed herein was designed to address a real-world time prediction problem faced by an Information Technology (IT) organization. In this organization, the incident management process is supported by the ServiceNow^{TM} platform, which enables extraction of the event log and a series of descriptive incident attributes. Because it is an applied experiment, there is no prior initiative for comparison. To overcome this problem, the selection of attributes performed by human experts was used as the baseline. The semantic reasonability of the selected attributes in each case was analyzed in this real-world incident management process. The results show that only the wrapper-based solution could outperform human experts.

In summary, our goal is to discover an attribute subset that allows generating a model capable of minimizing the prediction error of the incident completion time during its resolution process. Fig. 1 presents an overview of the proposed strategy. The top of the figure shows the sequence of actions followed to build an enriched event log used to build the prediction models. The remaining part of the figure shows the three attribute selection methods explored in this paper: *(i)* expert-driven selection [used herein as our baseline for comparison], *(ii)* the filter with ranking and *(iii)* wrappers with two search techniques – hill-climbing and best-first.

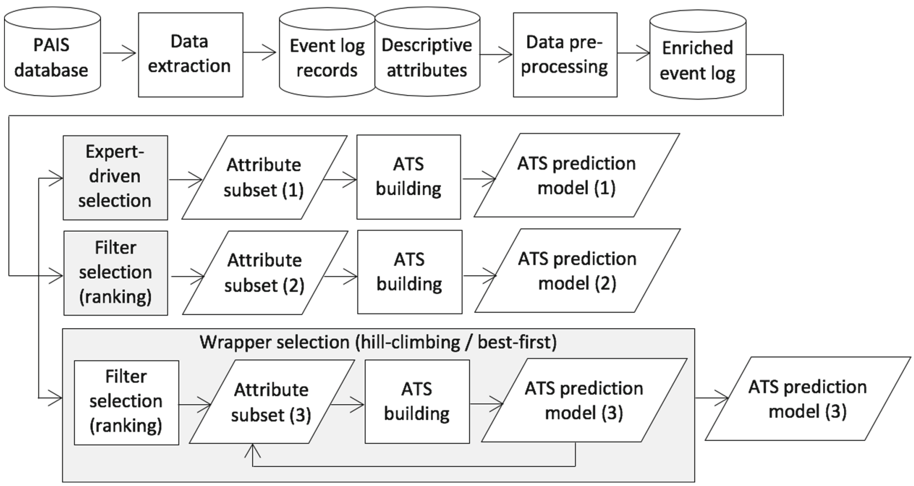

Fig. 1. Proposed strategy overview

The contribution of this work is threefold:

1. We present the feasibility of an automatic attribute selection approach used to improve the performance of prediction models that are sensitive to these attributes.
2. We confirm through experimental results that automatic methods can outperform human experts for a real-world incident management context even considering the own specific characteristics of such a context.
3. We provide the dataset used in our experiment, containing an event log enriched with data loaded from a relational database underlying the related PAIS, which can be used for replicability or other experiments.

The remainder of this paper shows: an overview of concepts related to attribute selection and annotated transition systems and some related work; the research method for experimentation, including the strategies for attribute selection, the application domain and the event log used; the findings of the experiments conducted; the discussion of such findings; and finally the conclusions.

2 Literature Review and Theoretical Background

This section presents the main concepts related to attribute selection and ATS as a theoretical basis for the rest of the paper and an analysis of the related works found in the literature review.

2.1 Attribute Selection

According to Blum and Langley [15], before undertaking automated learning activities, two tasks are needed to be carried out – deciding which features (or attributes) to use in describing the concept to be learned and deciding how to combine such features. Following this assumption, attribute selection is proposed herein as an essential phase to build prediction models capable of predicting completion time. The taxonomy of methods for selecting attributes typically uses three classes filters, wrappers and embedded [14]. A fourth class – heuristic search – is highlighted by Blum and Langley [15], however, one could say that this class is an extension of filter methods. In this paper, we apply the filter and wrapper methods [12,14,15], which are briefly described as follows:

- **Filter:** filter methods aim to select relevant attributes – those that alone or together can generate a better performing predictor than that generated from a set of irrelevant attributes – and remove irrelevant attributes. These methods are seen as a pre-processing step, seeing that they are applied independently and before the learning model chosen. Because of their independence, filter methods are often run-time competitive when compared to other attribute selection methods and can provide a generic attribute selection free from the behavior influence of learning models. In fact, using filters reduces the decision space dimensionality and has the potential to minimize the over-fitting problem. In this paper, a filter method based on correlation analysis is applied. Each attribute is individually evaluated based on its correlation with the target attribute (i.e., the instance completion time).

- **Wrapper:** in wrapper methods, the attribute selection is carried out through an interaction with an interface of the learning model, which is seen as a black box. There is indeed a space of search states (i.e., combinations of attributes) that needs to be explored using some search technique. Such a search is driven by the accuracy got with the application of the learning model in each search state, considering the parameters (or, in the case of this paper, the attributes) that characterize that search state. In this paper, we apply: two well-known search techniques – hill-climbing and best-first (described below); ATSs as the learning model (cf. Sect. 2.2); and Mean Absolute Percentage Error (MAPE) [16,17] as the metric to evaluate the learning model accuracy, defined as

$$MAPE = \frac{1}{n} * \sum_{t=1}^{n} \frac{|F_t - A_t|}{A_t},$$

where n is the number of events in the log, F_t is the result got with the predictor for each event of the log and A_t is the expected/known prediction value, which represents the remaining time to complete the process instance and is calculated from the time the event was logged in until the instance is completed.

Hill-climbing is one of the simplest search techniques; it expands the current state, creating new ones, moves to the next state with the best evaluation, and stops when no child improves the current state. Best-first search differs from hill-climbing as it does not stop when no child improves the current state; instead, the search attempts to expand the next node with the best evaluation in the open list [12].

2.2 Annotated Transition Systems

Using transition systems in process mining was proposed by Aalst *et al.* [18], as part of an approach to discovering control-flows from event logs. Then, transition systems were extended with annotations (given rise to ATS), whose aim is to add statistical characteristics of a business process. ATSs can be applied as a predictor of the completion time of a process instance based on the annotated statistical data [3]. According to the authors, ATSs include alternatives for state representation, allowing to address over-fitting and under-fitting, which are frequent in prediction tasks.

Briefly, a transition system is defined as the triplet (S, E, T), in which S is a space of states, E is a set of labeled events and T is the transition relation such that $T \subseteq S \times E \times S$. A state is an abstraction of k events in the event log, which have occurred in a finite sequence σ that is called 'trace'. σ is represented by a string of symbols derived using abstraction strategies. Five strategies are presented by Aalst *et al.* [18], from which the following two are applied in the experiments presented herein:

1. *Maximal horizon*, which determines how many events must be considered in the knowledge representation of a state.
2. *Representation*, which defines three ways to represent knowledge about past and future at a trace momentum, i.e., per:
   - *Sequence*, recording the order of activities in each state.
   - *Multiset*, ignoring the order and considers the number of times each activity is performed.
   - *Set*, considering only the presence of activities.

To create the ATS, each state is annotated taking information collected from all traces that have visited it [3]. For time analysis, for example, this annotation considers information about the completion time of the instances related to each earlier trace, i.e., the annotation is carried out in a supervised way. The information is aggregated in each state producing statistics such as average times, standard deviation, median times etc. Such annotations allow using ATSs as a predictor. Thus, predicting the completion time for a running trace referring to some process instance can be carried out from its current state in the ATS flow.

Berti [7] also applied ATS for prediction, however, with partial and weighted traces aiming at dealing with changes during the running process. The ATS was extended through machine learning and enriched with date/time information and probability of occurrence of activities in the traces, by Polato *et al.* [8]. As several factors influence prediction, the view on the need to deal with information that enriches the ATS context is also used in the approach addressed herein.

2.3 Related Work

Only Hinkka *et al.* [11] presented a strategy with a purpose similar to the one presented herein, i.e., choosing the attribute configuration of the input log for building the predictor. The approach of these authors extracts structural features from an event log (i.e., activity counting, transitions counting, occurrence ordering), submits them to a selection process, and then uses the features selected to describe process instances. These process instances are used to create categorical prediction models. Different feature selection methods were applied, based on randomness, statistics, heuristic search and clustering. Among the strategies used by the authors, recursive elimination – a wrapper method – was the best performing selection method (84% of accuracy); however, it was one of the most expensive in terms of time response. Despite the similarity, this work is not directly comparable with ours since these authors work with a simple binary classification scenario whereas we work with numerical prediction, i.e., a continuous scenario. Moreover, our strategy does not use recursive elimination as them as our search method is a simple forward selection.

Alternatively, Evermann, Rehse and Fettke [19] and Tax *et al.* [20] also worked with the choice of the configuration of the predictor input log, but implicitly and automatically when using deep learning. Prediction is done directly from the descriptions of process instances, i.e., no process model is used or discovered as a basis for prediction. As a disadvantage of this type of approach, it is hard to explain the reasonableness of the predictions made when considering the process context, i.e., the implicit extraction of features does not allow easily interpreting the information leading to the results of the prediction. As a result, this type of solution hinders the use of the selected attributes for process improvement purposes.

3 Research Method

This section details the proposed solution and the basis for the experiments.

3.1 Attribute Selection Strategies

An overview of the proposed strategy for attribute selection is presented in Fig. 1 and detailed in this section.

For the **first strategy** – the expert-driven selection, no standard procedure was followed, since it fully depends on human judgment. This judgment highly depends on the application domain, among other factors. In the next section, the rationale specifically followed for the case used in our experiment is presented.

For the **second strategy** – the filter with ranking, stable concepts of specialized literature were followed [12,14,15]. Ranking was applied as pre-processing, as suggested by Kohavi and John [12], to create a baseline for attribute selection, regardless of the prediction model in use. The ranking should be created through a variance analysis by correlating the independent variables (i.e., the descriptive attributes) and the dependent variable (i.e., the prediction target attribute). Since most of the descriptive attributes are categorical in this context, the statistic η^2 (Eta squared) should be applied, as explained by Richardson [21]. From the ranking results, the filter method should be executed n times by combining the attributes as follows: $\{1^{st}\}; \{1^{st}, 2^{nd}\}; \ldots; \{1^{st}, 2^{nd}, \ldots, n^{th}\}$.

For the **third strategy** – the wrapper with hill-climbing and best-first [12], a forward selection mode[2] was applied. The search space is composed of all combinations of the attributes pre-selected by the filter with ranking strategy. Each one of the combinations represents a state in such a space, whose quality measure is calculated as the predictive power achieved by the predictor generated with the attribute subset associated with this model. For real problems, an exhaustive search procedure is probably unfeasible, and hence using heuristic search procedures is justified. Algorithms 1 and 2 show, respectively, how hill-climbing and best-first searches are carried out for our attribute selection strategy. The building function *build-ATS()* of an ATS and the evaluation function *eval()* of the ATS use, respectively, a training log excerpt (e_{train}) and a testing log excerpt (e_{test}), which represent disjoint subsets of the original event log (e) generated in

[2] In the forward selection, the search initial point is a singleton attribute subset to which one new attribute is incorporated at each new step in the search.

Algorithm 1. Hill-climbing technique

input: set of attributes l, event log e;
output: canonical subset of attributes l_{final};

$l_{final} \leftarrow \emptyset$;
$\text{ATS}_{best} \leftarrow \emptyset$;
repeat
  $l_{expand} \leftarrow l - l_{final}$;
  $\text{ATS} \leftarrow \emptyset$;
  for $i = 1$ to $len(l_{expand})$ **do** ▷ State expansion
    $\textit{att-set}[i] \leftarrow concat\ (l_{final},\ l_{expand}[i])$;
    $\text{ATS}[i] \leftarrow \textit{build-ATS}(\textit{att-set}[i],\ e_{train})$;
  $i_{best} \leftarrow \textit{arg-min}\ (eval\ (\text{ATS},\ e_{test}))$;
  if $(eval(\text{ATS}_{best}, e_{test}) > eval(\text{ATS}[i_{best}], e_{test}))$ **then**
    $\text{ATS}_{best} \leftarrow \text{ATS}[i_{best}]$;
    $l_{final} \leftarrow \textit{att-set}[i_{best}]$;
until $(l_{final} \neq \textit{att-set}[i_{best}])$ **or** $(l_{expand} = \emptyset)$
return l_{final}

the cross-validation procedure. The function *eval()* returns the MAPE for the ATS under evaluation and is used for a single ATS and a set of ATSs. The minimization function, *arg-min()*, applied to the ATS evaluation, returns the index of the model that produces the lowest MAPE when applied to the testing log. In Algorithm 2, there are two lists (open and closed) that maintain the states that represent the sets of attributes generated by the search and are used by the function *build-ATS()* to create the ATSs related to each state under evaluation. The search is interrupted when the maximum expansion counter is achieved.

For all selection methods, ATS is applied as the prediction model responsible for generating the estimates of the incident completion times, including to act as a state evaluator in the wrapper search spaces. For practical purposes, the ATS can be generated from an attribute subset which properly describes the currently completed incidents. From this point, ATS can be applied to predict the completion time of new incidents at run-time.

3.2 Application Domain

Operating areas in organizations are often complex, requiring a constant search for optimization to become more stable and predictable. In IT, this optimization is sought by adopting good practice frameworks such as the Information Technology Infrastructure Library (ITIL) [22]. ITIL covers several IT service management processes, from which incident management is the most commonly used one [23]. The incident management process addresses actions to correct failures and restore the normal operation of a service, as soon as possible, to minimize the impact on business operations [22]. Systematizing this business process allows defining monitoring indicators, including the completion time for

Algorithm 2. Best-first technique

input: set of attributes l, event log e, maximum # expansion movements with no improvement $max\_expcount$;
output: canonical subset of attributes l_{final};

$l_{final} \leftarrow \emptyset$;
$l_{closed-states} \leftarrow \emptyset$;
$l_{open-states} \leftarrow$ *expand-state*($\emptyset, l_{closed-states}, l$);
$\text{ATS}_{best} \leftarrow \emptyset$;
repeat
    ATS $\leftarrow$ *build-ATS*($l_{open-states}$, e_{train});
    $i_{best} \leftarrow$ *arg-min* (*eval* (ATS, e_{test}));
    $currentstate \leftarrow l_{open-states}[i_{best}]$;
    $l_{open-states} \leftarrow l_{open-states} - currentstate$;
    $l_{closed-states} \leftarrow l_{closed-states} + currentstate$;
    if $(eval(\text{ATS}_{best}, e_{test}) > eval(\text{ATS}[i_{best}], e_{test}))$ **then**
        $\text{ATS}_{best} \leftarrow \text{ATS}[i_{best}]$;
        $l_{final} \leftarrow$ *att-set*($currentstate$);
        $expcount \leftarrow 0$;
    else
        inc($expcount$);
    $l_{expand} \leftarrow$ *expand-state*($currentstate, l_{closed-states}, l$);
    $l_{open-states} \leftarrow$ *concat*($l_{open-states}, l_{expand}$);
until $(expcount \leq max\_expcount)$ **or** $(l_{open-states} = \emptyset)$
return l_{final}

incident resolution (also known as 'ticket completion time'), one of the most important indicators for this process [23].

When an incident occurs, it is identified and reported by a caller. Afterward, a primary expectation is to know the incident completion time. The usual estimates follow ITIL best practices, which are based on some specific incident attributes like urgency, category etc. This approach is general and inaccurate since it aggregates many situations and common target completion times. As the process evolves from the identification and classification stage to the initial support, investigation and diagnosis, some attributes are updated, and new ones are added. This can usually lead to a number close to 100 attributes, depending on the scope of the system implementation. Considering this whole scenario, there is an open issue related to providing assertive estimates on incident completion time that is not adequately solved by simple statistical methods. Incident management systems commonly store descriptive information of process instances and audit information about the history of updates of the process in progress. Combining both types of information allows executing a detailed step-by-step process evaluation and hence deriving estimates for each recorded event.

ServiceNow$^{\text{TM}}$ is a proprietary platform in which IT process management is implemented regarding the ITIL framework. In this platform, the incident process management involves three actors in five basic process steps. The actors are: *caller*, affected by the unavailability or degradation of a service, caused by an incident; *service desk analyst*, responsible by registering and validating the data provided by the caller and executing the initial procedures to treat the incident; and *support analysts*, the group of agents responsible for further analyzing the incident and its causes and proposing workaround solutions to be applied until the service is reestablished or definitive solutions are found. The five basic process steps are: incident identification and classification, initial support, investigation and diagnosis, resolution and reestablishment, and closing.

3.3 Enriched Event Log

An enriched event log of the incident management process was extracted from an instance of the ServiceNow^{TM} platform used by an IT company[3]. Information was anonymized for privacy reasons. This enriched event log is composed of data gathered from both the audit system and the platform's relational database:

- **Event log records:** ServiceNow^{TM} offers an audit system that records data referring to events related to all data maintained by the system, including incident-related data. The main data recorded are event identifier, old data value, new data value, update timestamp and responsible user. Audit data was used to generate the main structure of the event log records to be mined. We considered 12 months (Mar-2016 to Feb-2017), totaling 24,918 traces and 141,712 events. Pre-processing was used to filter out the noise and organize audit records in an orderly sequence compatible with an event log format. Two audit log attributes were derived from this audit system *sys_updated_at* and *sys_updated_by*.
- **Incident descriptive attributes:** ServiceNow^{TM} has 91 incident descriptive attributes. Some are worthless for process mining, have missing or inconsistent data, or represent unstructured information (i.e., text), whose use is outside our scope. After removing such unnecessary attributes, the final set of descriptive attributes comprised 34 attributes (27 categorical, 3 numeric and 4 timestamp ones). These attributes include the attribute *closed_at*, which is used as the basis for calculating the dependent variable for prediction.

An excerpt from the enriched event log is shown in Table 1. It refers to one incident (INC001) and contains: one audit attribute (*sys_updated_at*) and the other four are descriptive attributes (*number*, *incident_state*, *category* and *assignment_group*).

Statistical data on the enriched event log is presented in Table 2. A well-defined behavior for the incident management process is observed, as most incidents (75%) go through up seven updates, 50% go through up five updates and on average six updates are needed to the total of incidents. There are some outliers, with 58 as the maximum number of updates for one incident. Regarding time (in days), the behavior resembles an exponential distribution.

[3] Available at http://each.uspnet.usp.br/sarajane/?page_id=12.

Table 1. Incident enriched event log excerpt

Number	*incident_state*	*sys_updated_at*	*Category*	*assig._group*
INC001	New	3/2/2016 04:57	Internet	Field service
	New	3/2/2016 16:52	Internet	Field service
	Active	3/2/2016 18:13	Internet	Field service
	Active	3/2/2016 19:14	Internet	Field service
	Awaiting UI	3/2/2016 19:15	Internet	Field service
	Awaiting UI	3/3/2016 11:24	Internet	Field service
	Awaiting UI	3/3/2016 12:33	Internet	Field service
	Awaiting UI	3/3/2016 12:43	Internet	Field service
	Active	3/3/2016 12:43	Internet	Field service
	Active	3/3/2016 12:54	Internet	Field service
	Active	3/3/2016 12:57	Internet	Inf. security
	Active	3/3/2016 13:14	Internet	Inf. security
	Active	3/3/2016 13:16	Internet	Service desk
	Active	3/3/2016 19:57	Internet	Field service
	Active	3/4/2016 10:56	Internet	Field service
	Resolved	3/4/2016 11:02	Internet	Field service
	Closed	3/9/2016 12:00	Internet	Field service

Table 2. Enriched event log statistics: per incident/day

	1^{st}Q.	2^{nd}Q.	3^{rd}Q.	Max	Mean	St. dev.
Per incident	3	5	7	58	6	3.67
Per day	0.01	0.40	5.29	336.21	6.67	21.20

4 Research Findings

This section presents the results of the experiments. The incident management process was used as the application domain. The enriched event log was split into 5 folds (i.e., 5 sublogs) to allow cross-validation on the ATS prediction models. The ATS accuracy is given in terms of the mean and the median MAPE [16] of the incident completion time taking all incidents in the test fold that are passing through the ATS states. Sojourn time is also considered. The ATS completeness (or non-fitting) was evaluated by accounting how many records do not have a corresponding state in the ATS. As a baseline for comparison, a prediction model based on human expertise-knowledge was first created.

Three experiments were conducted as described in Sect. 3. A set of ATSs was generated according to these parameter configurations:

- **Enriched event log:** the enriched event log was sampled by randomly creating two subsets, one with 8,000 (A) and another with 24,000 (B) incidents – with $A \subset B$.
- **Maximum horizon:** 1, 3, 5, 6, 7 and 'infinite' were used. The value 1 explores the simpler case with only the last event per incident trace; 3, 5, 6 and 7 explore the most frequent behaviors in this incident management process according to the statistics 'by incident' reported in Table 2; and, 'infinite' explores all events per incident trace.
- **State representation:** the three options described in Sect. 2.2 were used, i.e., *set*, *multiset* and *sequence* [18].

4.1 Experiment #1 – Expert-Driven Selection

First, attribute selection was driven by information about the domain held by human experts. According to ITIL best practices, to start the incident management, the caller should provide the initial incident information, which is complemented by the service desk agent, with information related to the incident category and priority (defined by impact and urgency). Additional information (attachments and textual descriptions) is also provided to help the support agents who need to act on the next stage, which is out of the scope of this work. Based on these practices, *incident_state*, *category* and *priority* were considered the most adequate attributes to correctly define the process model in ATS: *incident_state* reports the stage at which the incident is; *category* shows the type of service the incident belongs to; and *priority* determines the focus requested by the business. For this scenario, 18 ATSs were generated and used as completion time predictor, for the enriched event log sample with 24,000 incidents, varying the horizon and state representation parameters. The results are shown in Table 3. The best results were got with horizon 3 and state representation *sequence*.

Table 3. Experiment #1 – average prediction results. Used attributes: *incident_state*, *category* and *priority*. Log sample: 24,000 incidents. Metric: MAPE (Mean and Median). NF = % of non-fitting incidents. Bold: best results.

Max Hor	Set			Multiset			Sequence		
	Mean	*Med*	*NF*	*Mean*	*Med*	*NF*	*Mean*	*Med*	*NF*
1	113.93	88.29	0.22	113.93	88.29	0.22	113.93	88.29	0.22
3	106.93	77.46	0.98	91.35	75.87	1.23	**72.36**	**63.66**	**1.38**
5	119.18	109.28	1.64	177.05	162.08	2.95	126.12	104.67	3.38
6	183.52	115.59	1.83	122.54	98.74	3.72	102.73	84.01	4.41
7	93.22	75.11	1.95	1190.87	1184.75	4.44	107.58	98.04	5.48
Inf.	1146.57	1123.24	2.31	92.12	75.21	8.03	88.32	72.98	9.00

Table 4. The 15 descriptive attributes with the highest correlation with the dependent variable and respective η values. Attribute descriptions are provided in the appendix.

Order	Attribute	η	Order	Attribute	η
1^{st}	Caller	0.54	9^{th}	Active	0.25
2^{nd}	Assigned_to	0.37	10^{th}	Priority_confirmation	0.24
3^{rd}	Assignment_group	0.35	11^{th}	Created_by	0.21
4^{th}	Symptom	0.33	12^{th}	open_by	0.20
5^{th}	Sys_updated_by	0.33	13^{th}	Location	0.14
6^{th}	Incident_state	0.32	14^{th}	Made_SLA	0.14
7^{th}	Subcategory	0.32	15^{th}	Knowledge	0.12
8^{th}	Category	0.27			

4.2 Experiment #2 – Filter with Ranking

Second, attribute selection was driven by filter using a ranking strategy. Following the strategy presented in Sect. 3, 15 attributes with the highest correlation with the dependent variable (i.e., the prediction target attribute, based on the attribute *closed_at*) were selected to compose the ranking. The variance analysis was carried out on the entire enriched event log. The attributes and correlation scores are listed in Table 4. These results showed that the descriptive attributes with the highest correlation with the dependent variable are those related with associated resources of the incident management process. Considering the ranking results, the filter method was executed by combining the attributes as follows: $\{Caller(1^{st})\}$; $\{Caller(1^{st}), Assigned\_to(2^{nd})\}$; $\ldots$; $\{Caller(1^{st}), Assigned\_to(2^{nd}), \ldots, Knowledge(15^{th})\}$. For this scenario, 18 ATSs were generated for each attribute subset and used as completion time predictor, for the enriched event log sample with 8,000 incidents, varying the maximum horizon and the state representation parameters. The results for each attribute subset are shown in Table 5. The best results were got with horizon 1 and the subsets {*Caller, Assigned_to*} and {*Caller, Assigned_to, Assignment_group*}, regardless of the state representation.

As a second part of experiment #2, aiming to compare the prediction results got through the ATS models generated using these two best ranked attribute subsets with the results got in experiment #1, two new set of ATSs were generated using as attributes those of best results in Table 5; however, using in this case the enriched event log sample with 24,000 incidents. The results are shown in Table 6. The results with the ranked attribute subsets were slightly worse than those got in experiment #1. By checking these results, one can notice that resource-related attributes often impair generating the prediction model, i.e., such attributes do not reflect the process behavior with the same fidelity that the control attribute do (i.e., the incident state). Regarding non-fitting, an explanation for the poor results could be the frequent changes in the values of the human resource assigned to solve different incidents.

Table 5. Experiment #2 – average prediction results. Used attributes: selected by filter. Log sample: 8,000 incidents. Metric: MAPE (Mean and Median). NF = % of non-fitting incidents. Bold: best results.

Att	Max Hor	Set			Multiset			Sequence		
		Mean	*Med*	*NF*	*Mean*	*Med*	*NF*	*Mean*	*Med*	*NF*
1	Inf.	160.22	140.99	20.77	114.62	109.79	30.95	114.62	109.79	30.95
2	**1**	**110.98**	**90.81**	**59.89**	**110.98**	**90.81**	**59.89**	**110.98**	**90.81**	**59.89**
3	**1**	**112.27**	**88.99**	**63.92**	**112.27**	**88.99**	**63.92**	**112.27**	**88.99**	**63.92**
4	6	129.41	98.90	72.22	123.72	96.08	72.72	122.83	95.11	72.73
5	4	128.71	98.52	72.89	128.36	98.11	73.08	128.49	98.15	73.08
6	Inf.	129.25	100.28	73.39	133.72	102.29	73.51	133.72	102.29	73.51
7	Inf.	146.08	117.20	73.58	129.63	98.36	73.70	129.63	98.36	73.70
8	Inf.	143.84	114.87	73.66	129.42	98.06	73.77	129.42	98.06	73.77
9	Inf.	143.84	114.87	73.66	129.42	98.06	73.77	129.42	98.06	73.77
10	Inf.	130.46	101.07	73.67	133.72	101.61	73.72	139.35	107.19	73.72
11	3	135.57	103.93	73.65	133.30	101.25	73.67	134.97	102.96	73.67
12	Inf.	147.31	118.41	73.76	130.57	99.36	73.86	130.57	99.36	73.86
13	7	127.16	97.58	73.78	128.37	98.20	73.87	128.28	98.16	73.87
14	Inf.	124.96	96.09	73.78	126.14	96.85	73.88	126.14	96.85	73.88
15	Inf.	125.70	96.75	73.78	130.25	98.98	73.88	130.25	98.98	73.88

Table 6. Experiment #2 – average prediction results. Used attributes: best attribute subsets selected by filter. Log sample: 24,000 incidents. Metric: MAPE (Mean and Median). NF = % of non-fitting incidents. Bold: best results.

Max Hor	Set			Multiset			Sequence		
	Mean	*Med*	*NF*	*Mean*	*Med*	*NF*	*Mean*	*Med*	*NF*
Attribute subset: {*caller, assigned_to*}									
1	208.61	196.42	30.10	208.61	196.42	30.10	208.61	196.42	30.10
3	102.09	89.17	32.48	86.41	72.50	33.87	98.69	84.37	33.90
5	90.73	76.30	33.31	**69.69**	**57.85**	**35.67**	80.97	69.10	35.73
6	292.51	280.42	33.44	77.53	65.66	36.15	82.78	70.92	36.20
7	171.55	159.95	33.51	91.22	79.66	36.41	103.14	90.27	36.46
Inf.	249.06	238.05	33.60	96.66	85.85	36.73	78.82	67.97	36.76
Attribute subset: {*caller, assigned_to, assignment_group*}									
1	80.17	67.87	34.04	80.17	67.87	34.04	80.17	67.87	34.04
3	93.16	80.65	37.48	102.64	86.15	38.58	131.73	118.08	38.67
5	91.34	80.96	39.22	**76.21**	**64.98**	**40.67**	86.20	74.89	40.75
6	85.55	74.76	39.58	94.38	83.01	41.04	78.05	66.67	41.11
7	96.99	85.00	39.76	102.01	86.35	41.19	105.66	94.33	41.25
Inf.	85.96	74.00	40.03	81.33	70.36	41.33	79.76	68.76	41.36

Table 7. Experiment #3 – average prediction results. Used attributes: Best attributes selected by wrapper (*incident_state, location*). Log sample: 8,000 incidents. Metric: MAPE (Mean and Median). NF = % of non-fitting incidents. Bold: best results.

Max Hor	Set			Multiset			Sequence		
	Mean	*Med*	*NF*	*Mean*	*Med*	*NF*	*Mean*	*Med*	*NF*
1	501.18	450.23	0.88	501.18	450.23	0.88	501.18	450.23	0.88
3	528.98	522.63	1.92	497.56	475.72	2.70	92.71	64.01	2.96
5	185.12	66.39	2.51	113.64	84.77	5.71	143.45	72.07	6.60
6	33.90	19.51	2.58	43.02	23.74	6.91	33.85	22.87	8.19
7	**17.82**	**10.13**	**2.69**	21.36	15.19	8.07	25.07	15.46	9.74
Inf.	60.69	42.95	2.92	251.79	230.73	14.01	239.53	218.17	15.50

4.3 Experiment #3 – Wrappers with Hill-Climbing and Best-First

Last, the attribute selection was driven by the wrapper method using a forward selection mode with the hill-climbing and best-first search techniques [12] (cf. Sect. 3). The search space is composed of all combinations of the 15 attributes pre-selected by the filter with ranking strategy, i.e., the attributes in Table 4. Thus, the search space had $2^{15} = 32,768$ states, taking the 18 ATSs generated for each state, the range of the horizon and the state representation parameters. As stated before, using heuristic search procedures is justified in this case. The wrapper method was carried out on the enriched event log sample with 8,000 incidents. For the best-first search technique, the maximum number of expansion movements with no improvement was set to 15. The prediction results for the ATSs generated for this scenario are listed in Table 7. Both search techniques resulted in selecting the same best attribute subset, which are {*incident_state, location*}. Despite the high agreement between the two search techniques, some information can be extracted from their execution processes:

- **Hill-climbing:** the stopping criterion was reached after the third expansion movement; 42 states of the search space were explored; the mean and median for all ATSs generated in the state representation *set* were on average 146.80 and 103.76, respectively; and the average for non-fitting was 8.97.
- **Best-first:** 17 expansion movements were done; 172 states of the search space were explored; in average, the mean and median statistics for all ATSs generated in the state representation *set* were 114.96 and 89.68, respectively; the average for non-fitting was 36.27.

The best results were got with horizon 7 and the state representation *set*; however, the results got with the other state representations for the same horizon are good as well. These results are significantly better than those results got by the filter and better in terms of mean and median than those got by the expert-driven selection. Overall, the low non-fitting results are promising.

As a second part of experiment #3, with the purpose of comparing the prediction results got with the ATS models generated with these attribute subsets selected by wrapper with the results got in experiments #1 and #2, a new set of ATSs was generated using as parameters those of best results in Table 7, however using now the enriched event log sample with 24,000 incidents. The results are shown in Table 8 and it is noticed that the best results (maximum horizon set to 5) overcome the best results got in the previous experiments considering the MAPE evaluations. The results for MAPE are less than half of those measures got by expert-driven selection keeping non-fitting values at the lowest level.

4.4 Summarized View

Table 9 shows information detailing the average number of states on each set of ATSs created in experiment instances. One can check that best results (experiments #1 and #3) for MAPE also have the small number of states when compared with experiment #2.

Table 8. Experiment #3 – average prediction results. Used attributes: best attribute subsets selected by wrapper. Log sample: 24,000 incidents. Metric: MAPE (Mean and Median). NF = % of non-fitting incidents. Bold: best results.

Max Hor	Set			Multiset			Sequence		
	Mean	*Med*	*NF*	*Mean*	*Med*	*NF*	*Mean*	*Med*	*NF*
1	138.60	97.59	0.35	138.60	97.59	0.35	138.60	97.59	0.35
3	107.69	52.48	0.85	69.02	47.17	1.09	65.57	37.25	1.22
5	**50.45**	**24.49**	**1.11**	**41.90**	**29.35**	**2.30**	**35.09**	**27.28**	**2.74**
6	69.32	48.98	1.13	59.71	52.16	2.95	57.13	47.21	3.57
7	132.81	110.51	1.16	153.96	114.83	3.57	68.53	56.39	4.36
Inf.	66.75	46.16	1.24	43.02	35.86	6.51	70.54	38.26	7.43

Table 9. Consolidated view of the numbers of ATSs' states. Log sample: 24,000 incidents. Metrics: AVGS = AVErage number of States in the ATS; SD = Standard Deviation. Bold: refers to the set of ATSs with the best performances.

Max Hor	Set		Multiset		Sequence	
	AVGS	*SD*	*AVGS*	*SD*	*AVGS*	*SD*
Attribute subset: {*incident_state, category, priority*} – Exp. #1						
1	648.8	3.11	648.8	3.11	648.8	3.11
3	2428.0	13.01	3734.0	24.44	**4139.6**	**28.89**
5	3273.0	16.30	7895.4	50.66	9071.2	57.39
6	3456.8	15.27	9853.2	63.19	11501.4	65.77
7	3535.0	14.10	11678.0	73.84	13711.4	65.17
Inf.	3660.6	20.69	18423.4	115.99	19653.4	84.88
Attribute subset: {*caller, assigned_to*} – Exp. #2						
1	15297.4	67.56	15297.4	67.56	15297.4	67.56
3	18814.4	92.89	40474.4	142.09	41558.0	147
5	17658.6	76.79	**52537.2**	**150.95**	54034.4	157.18
6	17343.0	72.94	56218.4	160.01	57671.0	166.91
7	17062.6	69.26	58919.6	169.46	60287.6	173.83
Inf.	16205.6	62.87	66131.8	171.75	66151.2	170.66
Attribute subset: {*caller, assigned_to, assignment_group*} – Exp. #2						
1	24305.6	58.72	24305.6	58.72	24305.6	58.72
3	34664.4	68.94	54123.0	135.33	55942.2	141.48
5	31740.2	65.22	**64243.8**	**166.85**	66125.2	173.65
6	30425.0	70.68	66610.8	175.79	68193.4	189.58
7	29282.6	58.83	68037.2	179.03	69336.0	191.92
Inf.	26093.6	49.53	70795.0	197.57	70820.6	201.09
Attribute subset: {*incident_state, location*} – Exp. #3						
1	901.2	14.13	901.2	14.13	901.2	14.13
3	2322.6	25.16	3586.0	34.62	3939.0	36.61
5	**2675.2**	**27.10**	**6950.4**	**44.95**	**7972.6**	**52.66**
6	2697.4	25.54	8481.2	41.61	9881.6	47.67
7	2706.2	27.88	9838.8	36.38	11590.0	39.82
Inf.	2634.8	26.03	15901.8	78.16	17259.6	69.63

5 Discussion of Findings

With the analysis of the results, we could verify that the expert-driven and the filter with ranking strategies allow us building models with similar predictive power. However, when checking the model fitting capabilities, some differences (1.38 and 35.67, respectively) are observed between them for the best results.

Such differences were caused because of the different process perspectives represented by the attribute subset used in each case. For the first case, the ATS generation was driven by incident descriptive attributes recommended by ITIL best practices suggested by human experts for incident clustering and routing; then, the resulting model could accurately represent the process. For the second case, the set of attributes automatically selected to build the ATS represents organizational and resource perspectives of the incident management process; what means that, in this case, the ATS captured how teams (i.e., people) act to support user requests and became highly specialized and incapable of generalizing the real process behavior. This happens because the attributes selected represent information that presumably changes frequently ('caller' and 'technical people' in charge of the incident). The MAPE results for experiment #1 were compared to those got for experiment #2, using the paired Wilcoxon test. This test showed that there is no statistical difference among the distributions of the MAPE values as with $p_{value} = 0.3125$ the null hypothesis for equal distributions cannot be rejected.

The wrapper-based experiment achieved an average MAPE measure (24.49) that is 38.47% of the average MAPE achieved in the expert-driven experiment. The model non-fitting continued in an even lowest level (1.11%) as that got in the first one. The paired Wilcoxon statistical test was applied to compare the MAPE results got for experiment #1 with those got for experiment #3. The null hypothesis for equal distributions was rejected with $p_{value} = 0.0312$. This result allows affirming that the attribute selection got with the wrapper is better than the expert's choice in terms of accuracy and generalization (i.e., low non-fitting) in this incident management process.

The attribute subset selected by wrapper unifies expert knowledge with an organizational perspective, which produced a completion time predictor with high accuracy and low non-fitting rates. The results were similar for hill-climbing and best-first search techniques. This behavior has already been observed in experiments executed by Kohavi and John [12], in which, for diverse types of datasets, additional search effort did not produce better results.

6 Conclusions

Using the wrapper method could select a set of attributes that supported a significant improvement in the accuracy of ATS as a prediction model when compared to both the filter and the expert knowledge. Furthermore, such a search process points out that the maximum horizon and distinct types of state representations have a high influence on the prediction model results. This approach has the potential as a useful pre-processing step before applying other prediction methods besides the ATS method used in this study. These results are important when considering a business process scenario in which different actors need to collaborate for its execution, generating complexity and unpredictability of the completion time, for example.

This paper focuses on a specific application domain to illustrate that the proposed strategy is a way to solve a generic problem. However, while it is a promising heuristic procedure, there is no guarantee that the search will yield satisfactory results for all applications in similar scenarios. As the proposed approach performs a search in the event log derived from a specific process, it implements an inductive reasoning mechanism dependent on the properties of such an underlying process, regardless of the chosen prediction technique. As a result, for each specific case of application, different results are likely to be got.

In addition, it is still necessary to verify the influence of outliers throughout the process (search and prediction) as the results got in the experiments presented some varying degree. Using other search methods (such as genetic algorithms) or other options to build process model-based predictors (such as Petri nets or variations of ATSs), applied on benchmark event logs for comparison, are points for exploration.

Acknowledgments. This work was funded by the São Paulo Research Foundation (Fapesp), Brazil; grants 2017/26491-1 and 2017/26487-4.

Appendix

A brief description of the 15 attributes listed in Table 4 is presented in Table 10.

Table 10. Description of the 15 attributes used in the experiment

ID	Attribute	Description
1	*caller*	Identifier of the user affected
2	*incident_state*	Eighth levels controlling the incident management process transitions from opening until closing the case
3	*assigned_to*	Identifier of the user in charge of the incident
4	*assignment_group*	Identifier of the support group in charge of the incident
5	*symptom*	Description of the user perception about the service availability
6	*sys_updated_by*	Identifier of the user who updated the incident and generated the current log record
7	*subcategory*	Second level description of the affected service (related to the first level description, i.e., to *category*)
8	*category*	First level description of the affected service
9	*active*	Boolean attribute indicating if the record is active or closed/canceled
10	*priority_confirmation*	Boolean attribute indicating whether the *priority* field has been double-checked
11	*created*	Incident creation date and time
12	*open_by*	Identifier of the user who reported the incident
13	*location*	Identifier of the location of the place affected
14	*made_SLA*	Boolean attribute that shows whether the incident exceeded the target SLA
15	*knowledge*	Boolean attribute that shows whether a knowledge base document was used to resolve the incident

References

1. de Leoni, M., van der Aalst, W.M., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. Inf. Syst. **56**, 235–257 (2016). https://doi.org/10.1016/j.is.2015.07.003
2. van der Aalst, W.M.P.: Process Mining - Discovery, Conformance and Enhancement of Business Processes, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-642-19345-3
3. van der Aalst, W., Schonenberg, M., Song, M.: Time prediction based on process mining. Inf. Syst. **36**(2), 450–475 (2011). https://doi.org/10.1016/j.is.2010.09.001
4. Boonjing, V., Pimchangthong, D.: Data mining for positive customer reaction to advertising in social media. In: Ziemba, E. (ed.) AITM/ISM-2017. LNBIP, vol. 311, pp. 83–95. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77721-4_5
5. Łobaziewicz, M.: The role of ICT solutions in the intelligent enterprise performance. In: Ziemba, E. (ed.) AITM/ISM-2016. LNBIP, vol. 277, pp. 120–136. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53076-5_7
6. Pawełoszek, I.: Data mining approach to assessment of the ERP system from the vendor's perspective. In: Ziemba, E. (ed.) Information Technology for Management. LNBIP, vol. 243, pp. 125–143. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30528-8_8
7. Berti, A.: Improving process mining prediction results in processes that change over time. In: Proceedings of the 5th International Conference on Data Analytics, pp. 37–42. IARIA (2016)
8. Polato, M., Sperduti, A., Burattin, A., de Leoni, M.: Data-aware remaining time prediction of business process instances. In: Proceedings of the 2014 International Joint Conference on Neural Networks, pp. 816–823. IEEE, July 2014. https://doi.org/10.1109/IJCNN.2014.6889360
9. Rogge-Solti, A., Vana, L., Mendling, J.: Time series Petri net models - enrichment and prediction. In: Proceedings of the 5th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA), pp. 109–123 (2015). https://doi.org/10.1007/978-3-319-53435-0_6
10. Rogge-Solti, A., Weske, M.: Prediction of business process durations using non-Markovian stochastic Petri nets. Inf. Syst. **54**, 1–14 (2015). https://doi.org/10.1016/j.is.2015.04.004
11. Hinkka, M., Lehto, T., Heljanko, K., Jung, A.: Structural feature selection for event logs. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 20–35. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_2
12. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**(1), 273–324 (1997). https://doi.org/10.1016/S0004-3702(97)00043-X
13. do Amaral, C.A.L., Fantinato, M., Peres, S.M.: Attribute selection with filter and wrapper: an application on incident management process. In: Proceedings of the 15th Conference on Advanced Information Technologies for Management (AITM) in Federated Conference on Computer Science and Information Systems (FedCSIS), vol. 15, pp. 679–682 (2018). https://doi.org/10.15439/2018F126
14. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003). https://doi.org/10.1162/153244303322753616
15. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artif. Intell. **97**(1–2), 245–271 (1997). https://doi.org/10.1016/S0004-3702(97)00063-5

16. Armstrong, J.S., Collopy, F.: Error measures for generalizing about forecasting methods: empirical comparisons. Int. J. Forecast. **8**(1), 69–80 (1992). https://doi.org/10.1016/0169-2070(92)90008-W
17. de Myttenaere, A., Golden, B., Grand, B.L., Rossi, F.: Mean absolute percentage error for regression models. Neurocomputing **192**, 38–48 (2016). https://doi.org/10.1016/j.neucom.2015.12.114
18. van der Aalst, W.M.P., Rubin, V., Verbeek, H.M.W., van Dongen, B.F., Kindler, E., Günther, C.W.: Process mining: a two-step approach to balance between underfitting and overfitting. Softw. Syst. Model. **9**(1) (2008). https://doi.org/10.1007/s10270-008-0106-z
19. Evermann, J., Rehse, J.R., Fettke, P.: Predicting process behaviour using deep learning. Decis. Support Syst. **100**, 129–140 (2017). https://doi.org/10.1016/j.dss.2017.04.003
20. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 477–492. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_30
21. Richardson, J.T.E.: Eta squared and partial eta squared as measures of effect size in educational research. Educ. Res. Rev. **6**(2), 135–147 (2011). https://doi.org/10.1016/j.edurev.2010.12.001
22. itSMF: Global survey on IT service management. http://www.itil.co.il
23. Marrone, M., Gacenga, F., Cater-Steel, A., Kolbe, L.: IT service management: a cross-national study of ITIL adoption. Commun. Assoc. Inf. Syst. **34**, 49.1–49.30 (2014). https://doi.org/10.17705/1CAIS.03449