Downloaded from orbit.dtu.dk on: Dec 02, 2023

DTU Library

# Behavioral Recommender System for Process Automation Steps

**Sani, Mohammadreza Fani; Nikraftar, Fatemeh; Sroka, Michal; Burattin, Andrea**

*Published in:*
Proceedings of the 12th International Conference on Data Science, Technology and Applications DATA

*Link to article, DOI:*
10.5220/0012060800003541

*Publication date:*
2023

*Document Version*
Peer reviewed version

Link back to DTU Orbit

*Citation (APA):*
Sani, M. F., Nikraftar, F., Sroka, M., & Burattin, A. (2023). Behavioral Recommender System for Process Automation Steps. In *Proceedings of the 12th International Conference on Data Science, Technology and Applications DATA* (Vol. 1, pp. 255-262). SciTePress. https://doi.org/10.5220/0012060800003541

**General rights**
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Behavioral Recommender System for Process Automation Steps

Mohammadreza Fani Sani[1[0000−0003−3152−2103]], Fatemeh Nikraftar[2], Michal Sroka[1[0000−0002−7505−2521]], and Andrea Burattin[2[0000−0002−0837−0183]]

[1] Microsoft Development Copenhagen, Denmark. {mfanisani,misroka}@microsoft.com
[2] Technical University of Denmark, Copenhagen, Denmark. andbur@dtu.dk

**Abstract.** Process automation is used to increase the performance of processes. One of the leading process automation tools is Microsoft Process Advisor. This tool requires users to select the corresponding *connectors* for the automation of different tasks, which can be a challenging endeavor for users who have limited business knowledge as there are various connectors and templates exist. To overcome this challenge, we present a process-aware recommender system for connectors that eases the labeling task for end users. The results of applying this method to real event logs indicate that it can recommend relevant connectors and, therefore, the usage of the same mechanism might be generalized to broader contexts.

**Keywords:** Process Mining · Process Automation · Microsoft Process Automation

## 1 Introduction

Process mining bridges data science and business process management. It consists of process discovery, conformance checking, and process enhancement. Process discovery generates a process model from event logs, conformance checking compares the logs and model, and process enhancement provides insights for improvement van der Aalst, 2016.

Gartner estimated that process enhancement will get more attention in the upcoming years[3]. In this sub-field, one of the research directions that helps process improvement is robotic process automation Leno et al., 2020. The goal of this field is to detect the bottlenecks of the process and try to automate the tasks that require more time. In this regard, we aim to detect and handle routine and administrative tasks automatically using current technologies. As a positive effect, we will reduce the resources' costs and the required times to handle the process instances. However, robotic process automation techniques also have their limits, and one of the fundamental challenges is selecting a suitable activity or process to automate, thus understanding which components should be involved in the automation. This task, in the literature, is referred to as task mining Syed et al., 2020.

One of the leading process automation tools is Microsoft Process Advisor (MPA)[4]. This tool provides the opportunity to automate manual and repetitive tasks that often take a lot of time. For example, in an accounting company, a resource should find the amount of a receipt and convert its currency by finding the rates in a browser and finally send the converted fee to the customer by email. Using MPA, it is possible to record the event log based on the execution of some process instances and provide the corresponding process automation steps. To provide such automation, the user needs to connect tasks to the corresponding connectors, i.e., proxies or wrappers around an API that lets the underlying service communicate with several Microsoft products. Currently, MPA supports more than 275 connectors and thousands of pre-built templates that allow for easy integration of popular end-services in workflow development/improvement. One of the challenges of automation in MPA is selecting the most suitable connector after recording the process. Without having knowledge of the process, the selection of proper connectors could be a challenging task.

To overcome this challenge, in this paper, we propose a recommender system that provides the best connectors for automating a process. To evaluate its accuracy, we develop it and apply it to 50 real process scenarios Sroka and Fani Sani, 2022. The results indicate that the proposed recommender system is able to help the end-user select the corresponding connector.

[3] See https://www.gartner.com/doc/reprints?id=1-28SA9BAA&ct=220118&st=sb
[4] See https://powerautomate.microsoft.com/en-us/process-advisor/

The structure of the remaining part of the paper is as follows. First, we provide related work in the area of process mining and recommender systems. Thereafter, the preliminaries that ease the rest of the paper. After that, in Section 4, we explain the proposed recommender system. Section 5 provides the results of evaluating the proposed method. Finally, in Section 6, we conclude the paper and propose some new directions to continue this research.

## 2 Related Work

Several works have been done in the areas of recommender systems and process automation in process mining and a good overview of these is available in Eili et al., 2021, where different systems are surveyed.

Different process discovery techniques exist, each employing various parameters that sometimes make it difficult which technique and setting should be used. There are some works, such as Ribeiro et al., 2014; Wang et al., 2012, that aim to help users to discover more suitable process models. In Jr. et al., 2021, the authors use a meta-learning approach that is able to recommend a suitable process discovery configuration with high accuracy. Moreover, the work presented in Schonenberg et al., 2008 discusses the possibility of applying process mining techniques to recommend to end-users what should be done in the next phase of the process. Furthermore, Seeliger et al., 2018 proposes an interactive recommender system that provides suggestions for visualising the process model. In Terragni and Hassani, 2018, the authors propose some possibilities to use process mining on the logs of users' interactions on websites to explore the customer journey, predict their future activities, and recommend actions that maximize particular performance indicators.

In Geyer-Klingeberg et al., 2018, the authors present a process mining technique to enable effective RPA activities towards process improvement. Moreover, Wanner et al., 2019 discusses the capabilities of processes-based techniques with RPA and proposes an automatable indicator system as well as RPA activities to maximize the automation investment.

In Mayr et al., 2022, a survey on task mining has been reported and several applications and challenges in front of this research are represented. Moreover, Choi et al., 2022 has proposed a tool to record the interactions with user interfaces and to generate an event log that can be used to bridge the gap between process mining and RPA by detecting the tasks that can be automated.

Some tools like MPA[4] provides templates/connectors to automate tasks and activities. We aim to provide a connector recommender system to help users in the selection of corresponding connectors.

## 3 Preliminaries

In this section, some process mining concepts are discussed. In process mining, we use events to provide insights into the execution of business processes. Each *event* is related to specific activities of the underlying process. Furthermore, we refer to a collection of events related to a specific process instance as a *case*. Both cases and events may have different attributes. An event log that is a collection of events and cases is defined as follows.

**Definition 1 (Event Log).** *Let $\mathcal{E}$ be the universe of events, $\mathcal{C}$ be the universe of cases, $\mathcal{AT}$ be the universe of attributes, and $\mathcal{U}$ be the universe of attribute values. Moreover, let $C \subseteq \mathcal{C}$ be a non-empty set of cases, let $E \subseteq \mathcal{E}$ be a non-empty set of events, and let $AT \subseteq \mathcal{AT}$ be a set of attributes. We define $(C, E, \pi_C, \pi_E)$ as an event log, where $\pi_C : C \times \mathcal{AT} \nrightarrow \mathcal{U}$ and $\pi_E : E \times \mathcal{AT} \nrightarrow \mathcal{U}$. Any event in the event log has a case, therefore, $\nexists_{e \in E}(\pi_E(e, case) \notin C)$ and $\bigcup\limits_{e \in E}(\pi_E(e, case)){=}C$.*

*Furthermore, let $\mathcal{A} \subseteq \mathcal{U}$ be the universe of activities and let $\mathcal{A}^*$ be the universe of sequences of activities. For any $e \in E$, function $\pi_E(e, activity) \in \mathcal{A}$, which means that any event in the event log has an activity. Moreover, for any $c \in C$ function $\pi_C(c, trace) \in \mathcal{E}^* \setminus \{\langle\rangle\}$ that means any case in the event log has a trace (i.e., a sequence of events). Having a trace of a case and the activities of the events in the trace, we can have a variant of a case, i.e., $\pi_C(c, variant) \in \mathcal{A}^* \setminus \{\langle\rangle\}$.*

Therefore, mandatory attributes for events are *case* and *activity*, and for cases are *trace* and *variant*.

There are different notations to show a process model, e.g., Petri net Petri and Reisig, 2008 and BPMN Chinosi and Trombetta, 2012 models. However, it is also possible to describe a process model by the complete sequence of activities that is possible to execute by it. In the following, we define a process model and a process discovery algorithm.

**Definition 2 (Process Model and Process Discovery).** *Let $\mathcal{A}$ be the universe of activities. $\mathcal{M} = P(A^*) \setminus \{\}$ is the universe of process models where $P(X)$ is the powerset of set $X$.*

*Moreover, let $\mathcal{EL}$ be the universe of event logs, $pd : \mathcal{EL} \rightarrow \mathcal{M}$ is a process discovery algorithm that returns a process model for each event log.*

To compute how a process model and an event log conform to each other, we use conformance checking measures. There are different conformance checking measures that are reported in the literature, e.g., alignments Adriansyah and Buijs, 2012 and token-replay Rozinat and van der Aalst, 2008. In the following, conformance checking is defined.

**Definition 3 (Conformance Checking).** *Let $\mathcal{A}$ be the universe of activities and let $\mathcal{M}$ denote the universe of process models. $cc$:$\mathcal{A}^* \times \mathcal{M} \rightarrow [0, 1]$ is a conformance checking function that returns how much behavior in a sequence of activities is represented in a process model.*

The conformance of a process model and an event log is the average conformance value of variants of all its traces ($\pi_C(c, variant)$). However, in this paper, we use the conformance checking at the variant level (i.e., we do not count repetitions of the same trace). The set of all variants in an event log $(C, E, \pi_C, \pi_E)$ is $\{\pi_C(c, variant)|c \in C\}$.

As explained, traces and variants are the sequences (of events and activities). We are able to use projection function on the sequence $\sigma = \langle x_1, x_2, \ldots, x_n\rangle$. So, if $Q \subseteq X$,$\upharpoonright_Q$: $X^* \rightarrow Q^*$ is a projection function that returns the concatenation of elements of the input sequence, i.e., $\sigma\upharpoonright_Q = \langle x_1\upharpoonright_Q \ldots x_2\upharpoonright_Q \ldots x_n\upharpoonright_Q\rangle$. For instance, $\langle a, b, b, c, d, e, f, h\rangle\upharpoonright_{\{a,b,h\}} = \langle a, b, b, h\rangle$. Note that, $x\upharpoonright_Q = \langle\rangle$ if $x \notin Q$.

## 4 Process Aware Recommender System

In this section, we explained the proposed process-aware connector recommender system. We first need to define a connector recommender system.

**Definition 4 (Connector Recommender).** *Let $\mathcal{E}$ denote the universe of activities and let $\mathcal{M}$ be the universe of process models. Moreover, let $\mathcal{L}$ be the universe of connectors. We define $cr : \mathcal{A}^* \times P(\mathcal{M}) \rightarrow P(\mathcal{L})$ as a* connector recommender *that receives a trace (i.e., a sequence of activities) and a set of process models, and returns a set of labels. Finally, $cr_k$ is a specific type of connector recommender that for trace $t \in \mathcal{A}^*$ and process models $\mathcal{M}_1 \subseteq \mathcal{M} \wedge |\mathcal{M}_1| \geq k$, we have $|cr_k(E, \mathcal{M}_1)| = k$.*

In other words, we recommend the connectors for the variants in the event log (can be extracted by $\pi_C(c, variant)$). The schematic view of this method is presented in Figure 1. The proposed method consists of two phases, i.e., *Train* and *Application*. In the training phase, as the proposed approach is a supervised method, we first need to provide the training/labeled event log(s). In other words, we need to have some traces labeled by their connectors.

It should be noted that one case or its trace can have one or more than one connectors. Therefore, a trace will be divided into some subsequences and each subsequence can relate to one connector. In the following, we define the labeled event log.

**Definition 5 (Labeling Event Log).** *Let $\mathcal{E}$ be the universe of events and let $\mathcal{L} \subset \mathcal{U}$ be the universe of labels. We define $\pi_E(e, label) \in \mathcal{L} \cup \tau$ that assigns a label or no value to each event.*

For each trace $\sigma$ in the labeled event log, we may have 0 to $|\sigma|$ labels. It should be noted that in reality, we usually assign labels to subsequences of a trace and we assign a label to all the events of a subsequence. Moreover, note that an activity can be labeled with different connectors in different events.

Afterward, we project the event log based on their assigned connectors, consequently, we will have one sub-log for each connector. We can use the following function for this purpose.

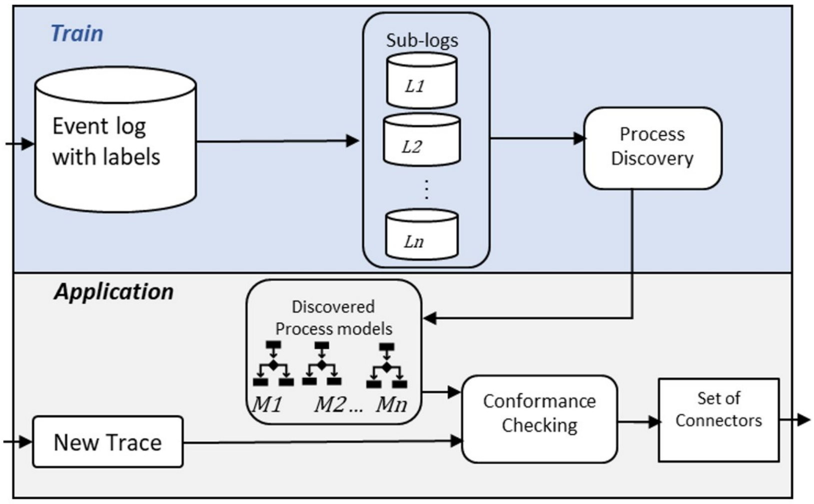


**Fig. 1.** A schematic view of the proposed connector recommender system. In the training phase, logs with the labels (i.e., the expected connector) are provided and a model is extracted for each label. The recommendation step (i.e., application) computes the conformance of the given trace against all models and selects the recommender associated with the highest conformance (i.e., the model that best fits the behavior of the recommender).

**Definition 6 (Event Log Projection).** *Let $\mathcal{EL}$ be the universe of event logs and let $EL{=}(C, E, \pi_C, \pi_E) \in \mathcal{EL}$ be a labeled event log. Moreover, let $L \in \mathcal{L}$ be a set of labels. We define $ep : \mathcal{EL} \times \mathcal{L} \nrightarrow P(\mathcal{EL})$ as an event log projection function that receives a labeled event log and a set of connectors and return a set of projected event logs where $ep(EL, L) = \{(C_l, E_l, \pi_C, \pi_E) | l \in L \wedge \big(E_c = \{e \in E | \pi_E(e, label) = l\} \wedge C_l = \{c \in C || \pi_C(c, trace) \upharpoonright E_c| > 0\}\}\big)\}$.*

In other words, for each connector, we gather all the events (and the cases if it remains at least one event in it) with the corresponding label in a sub-log. Note that $\bigcap\limits_{(C_l,E_l,\pi_C,\pi_E)\in ep(EL,L)} (E_l) = \{\}$; however, $\bigcup\limits_{(C_l,E_l,\pi_C,\pi_E)\in ep(EL,L)} (E_l) \leq |E|$ as it is possible that some of the events have no labels.

Thereafter, for each of the projected sub-logs, we discover a process model that represents the general behavior in that sub-log. It is not necessary to use similar parameters for different sub-logs. Note that as we define a process model as a possible sequence of activities, we can also consider the variants of the projected event logs (i.e., $M_l{=}\{\pi_C(c, variant) | c \in C_l\}$) as their process models. The discovered process models will be passed to the next phase. In other words, the output of the training phase is a set of process models for the connectors.

In the application phase, we aim to recommend some connectors for a given trace. For this purpose, we compute the conformance of the trace with all the discovered process models. The higher conformance values mean it is more likely that the trace relates to the corresponding connector. Thereafter, we recommend the top-$k$ connectors, i.e., the $k$ connectors with the highest conformance value.

## 5 Evaluation

In this section, we evaluate the proposed recommender system by applying it to real event logs. First, we explain the dataset that was used in the evaluation and the detail of the implementation and evaluation. Afterward, the evaluation results are discussed.

### 5.1 Experimental Setting

To evaluate the proposed connector recommender system, we used a dataset containing 50 processes, each of which contains 3 or more process instances (i.e., cases)[5]. We have labeled the event logs using

[5] The event log is available at https://github.com/microsoft/50BusinessAssignmentsLog

**Table 1.** Some information about the event logs that are used in the evaluation.

| Connector | Traces | Labeled Traces | Train Traces | Test Traces |
|---|---|---|---|---|
| *Approvals* | 30 | 30 | 23 | 7 |
| *Googlecalendar* | 30 | 26 | 20 | 6 |
| *Microsoftforms* | 21 | 21 | 16 | 5 |
| *Office365users* | 21 | 18 | 14 | 4 |
| *Onenote* | 24 | 15 | 11 | 4 |
| *Outlook* | 21 | 21 | 16 | 5 |
| *Planner* | 21 | 18 | 13 | 5 |
| *RSS* | 18 | 12 | 8 | 4 |
| *Sendmail* | 18 | 18 | 13 | 5 |
| *Sharepoint* | 18 | 15 | 11 | 4 |

experts with business knowledge, which gives us the training dataset to be used with the proposed method. There are 25 connectors (i.e., the labels) in the event log, however, we preprocess the dataset and keep only the 10 most frequent ones. We split the dataset into train and test parts: in the train event logs, we had 145 cases (on average 14.5 cases per connector); whereas in the test event logs, there were 49 cases. Some statistics of this event log is presented in Table 1[6].

We have implemented the proposed method in Python, using the PM4Py library Berti et al., 2019[7]. To discover process models, we have used the Inductive Miner Leemans et al., 2013 with the noise threshold value equal to 0.1 (to have more specific process models). Furthermore, to compute the conformance value, we have used the alignment technique. To recommend connectors, we use $cr_k$ with different $k$-values.

To evaluate the accuracy of the recommendation, we have used the following formula:

$$RetrieveRate = \frac{|\{LabeledConnector\} \cap \{RecommendedConnectors\}|}{|\{RecommendedConnectors\}|} \tag{1}$$

The higher value means a more accurate recommendation.

### 5.2 Experimental Results

As explained, we first need to discover process models using the projected event logs. Some information about the discovered process models is given in Table 2. In this table, activities indicate the number of labeled transitions in process models, and shared activities show how many times the activities appear in the other process models. The fitness and precision values indicate the quality of process models. As it can be noted in the table, for all process models, we have high fitness and low precision, which is mainly because of the noise threshold that is used (i.e., 0.1). Among the models, *RSS* has the fewest *activities* and *shared activities*. The highest $\frac{shared\ activities}{activities}$ value belongs to the process model of *Office365users* connector.

Fig. 2 shows the discovered process model of the *RSS* connector.

In the next step, we used the discovered process models to detect which connectors should be recommended for each trace in the test event logs. The results of using the proposed connector recommender system on the test data are presented in Table 3. The results show that for some connectors, e.g., *Office365users*, the labeled connector exists in the recommended connectors, even by recommending a few connectors. It is mainly because the discovered process model for these connectors is more distinguishable. On the other hand, for *RSS* connector, even by increasing the number of recommended connectors, the labeled connector does not exist in the list of recommendations. The low accuracy for

[6] This preprocessed labeled dataset is available at https://github.com/nikraftarf/Recommender-System-Based-on-processMining/tree/main/Data/Main-Data(Second-round).

[7] The implementation is available at https://bit.ly/3POX7z5

**Table 2.** Information about the discovered process models

| Model | Places | Arcs | Transitions | Activities | Shared Activities | Fitness | Precision |
|---|---|---|---|---|---|---|---|
| *Approvals* | 65 | 212 | 99 | 27 | 136 | 0.964 | 0.209 |
| *Googlecalendar* | 36 | 124 | 59 | 20 | 114 | 0.996 | 0.238 |
| *Microsoftforms* | 56 | 182 | 85 | 26 | 130 | 0.987 | 0.123 |
| *Office365users* | 48 | 160 | 76 | 19 | 123 | 0.960 | 0.154 |
| *Onenote* | 34 | 106 | 50 | 19 | 91 | 0.967 | 0.252 |
| *Outlook* | 59 | 196 | 93 | 26 | 128 | 0.992 | 0.199 |
| *Planner* | 58 | 192 | 92 | 23 | 104 | 0.979 | 0.124 |
| *RSS* | 27 | 86 | 43 | 13 | 74 | 0.987 | 0.353 |
| *Sendmail* | 49 | 152 | 70 | 19 | 119 | 1 | 0.272 |
| *Sharepoint* | 32 | 104 | 49 | 18 | 113 | 1 | 0.242 |

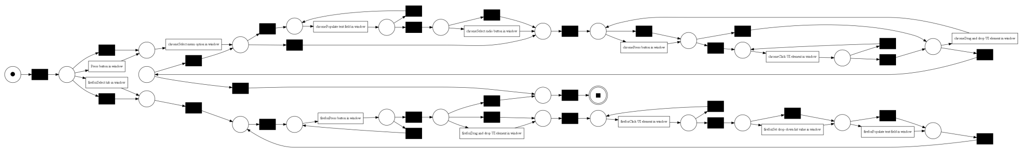

**Fig. 2.** The process model (with Petri net) discovered on projected event log of *RSS* connector.

the recommendations of *RSS*'s cases could be related to the few general labeled activities in its process model. Moreover, as we can see in Table 2, the precision of the process model of this connector is higher compared to the other process models.

It can be seen that by increasing the number of recommended connectors, i.e., $k$, we have a higher retrieve rate. However, recommending too many connectors can reduce the benefit of the proposed method. Therefore, finding optimal ways of properly configuring $k$ represents a significant future direction of research.

In Table 4, we have provided the frequency of different connectors when different k-values are used. Suppose that connector $c$ is recommended as the $m$th connector if we recommend $k \geq m$ connectors, this connector will be presented among them too. The results indicate that we recommend some connectors like *Sendmail* and *Microsoftforms* more often. Conversely, we infrequently recommend some connectors like *RSS* and *Onenote* which are recommended only 3 times. The reason for this is due to the absolute frequency of the activities in the projected event logs: some connectors are more common to appear in all logs hence these are able to replay traces of other connectors with higher fitness value.

### 5.3 Discussion and Limitation

While the work has been validated with experiments on real datasets, which revealed the efficacy of the technique, further investigations are needed to generalize the conclusions.

The proposed method assumes that most activities in the projected training event logs exist in the traces belonging to the connector we want to recommend. To overcome this limitation, NLP techniques such as semantic similarity methods Hayes and Henderson, 2021 can be applied to define equivalence classes for activities with different names but same action.

The performance of discovered process model is highly dependent on the quality of discovered process models. For example, if we discover a flower model van der Aalst et al., 2010, the process model will have high fitness values for different traces. The mining algorithm and the used setting for it can be seen as a hyperparameter of the technique and, like $k$, requires further investigation. But, contextual information is required to properly configure the technique.

**Table 3.** Retrieve Rate of the proposed method using different $k$ values for different connectors.

| | | **Retrieve Rate** | | | | |
|---|---|---|---|---|---|---|
| **Connector** | **Records** | $k = 1$ | $k = 2$ | $k = 3$ | $k = 4$ | $k = 5$ |
| *Approvals* | 7 | 0.57 | 0.71 | 0.71 | 1 | 1 |
| *Googlecalendar* | 6 | 0.5 | 0.5 | 0.67 | 1 | 1 |
| *Microsoftforms* | 5 | 0.6 | 0.6 | 0.8 | 1 | 1 |
| *Office365users* | 4 | 0.75 | 0.75 | 1 | 1 | 1 |
| *Onenote* | 4 | 0.25 | 0.25 | 0.5 | 0.5 | 0.5 |
| *Outlook* | 5 | 0.6 | 0.6 | 0.8 | 0.8 | 0.8 |
| *Planner* | 5 | 0.6 | 0.8 | 0.8 | 1 | 1 |
| *RSS* | 4 | 0 | 0 | 0 | 0 | 0.25 |
| *Sendmail* | 5 | 0.2 | 0.8 | 0.8 | 0.8 | 1 |
| *Sharepoint* | 5 | 0.4 | 0.4 | 0.6 | 0.8 | 0.8 |

**Table 4.** The number of times that each connector is recommended when different k-values are used.

| | *Approvals* | *Googlecalendar* | *Microsoftforms* | *Office365users* | *Onenote* | *Outlook* | *Planner* | *RSS* | *Sendmail* | *Sharepoint* |
|---|---|---|---|---|---|---|---|---|---|---|
| $k = 1$ | 3 | 3 | 5 | 6 | 1 | 9 | 9 | 0 | 9 | 2 |
| $k = 2$ | 8 | 5 | 12 | 9 | 2 | 12 | 16 | 0 | 20 | 5 |
| $k = 3$ | 11 | 6 | 24 | 15 | 2 | 14 | 20 | 1 | 25 | 7 |

The proposed method is a supervised technique and requires labeled event logs. Labeling can be expensive and time-consuming, but tools can be used to provide a rough estimate of the label. Additionally, labeling is only required once, and pre-trained models Qiu et al., 2020 can be used to reduce the effort. Transform learning and pre-trained models Qiu et al., 2020 can also be used due to the limited set of labels (i.e., the set of all connectors in MPA).

## 6 Conclusion

We proposed a process-aware recommender system that uses user behavior to suggest optimal steps for process automation. Results from applying the system to real event logs indicate that for most connectors, the correct recommendation is among the suggestions provided. This system can be used to improve the automation of processes and provide more efficient solutions.

To continue this research, we aim to consider other data attributes, such as resources, time dimensions, and other attributes not related to the control flow. We also plan to use association rule mining to understand multiple suggestions embedded in a single trace, and compare our technique with classical data mining approaches. This will help us to gain a better understanding of the process automation and provide more accurate recommendations. Furthermore, we aim to explore other techniques that could be used to improve the accuracy of the recommender system.

## References


Adriansyah, A., & Buijs, J. C. A. M. (2012). Mining process performance from event logs. *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, 217–218.

Berti, A., Van Zelst, S. J., & van der Aalst, W. (2019). Process mining for python (pm4py): Bridging the gap between process-and data science. *arXiv preprint arXiv:1905.06169*.

Chinosi, M., & Trombetta, A. (2012). BPMN: an introduction to the standard. *Comput. Stand. Interfaces*, *34*(1), 124–134.

Choi, D., R'bigui, H., & Cho, C. (2022). Enabling the gab between RPA and process mining: User interface interactions recorder. *IEEE Access*, *10*, 39604–39612.

Eili, M. Y., Rezaeenour, J., & Sani, M. F. (2021). A systematic literature review on process-aware recommender systems. *CoRR*, *abs/2103.16654*. https://arxiv.org/abs/2103.16654

Geyer-Klingeberg, J., Nakladal, J., Baldauf, F., & Veit, F. (2018). Process mining and robotic process automation: A perfect match. *Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018 co-located with 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018*, *2196*, 124–131.

Hayes, T. R., & Henderson, J. M. (2021). Looking for semantic similarity: What a vector-space model of semantics can tell us about attention in real-world scenes. *Psychological Science*, *32*(8).

Jr., S. B., Ceravolo, P., Damiani, E., & Tavares, G. M. (2021). Using meta-learning to recommend process discovery methods. *CoRR*, *abs/2103.12874*.

Leemans, S. J. J., Fahland, D., & van der Aalst, W. M. P. (2013). Discovering block-structured process models from event logs containing infrequent behaviour. *Business Process Management Workshops - BPM Beijing, China, August 26, 2013, Revised Papers*, *171*, 66–78.

Leno, V., Augusto, A., Dumas, M., Rosa, M. L., Maggi, F. M., & Polyvyanyy, A. (2020). Identifying candidate routines for robotic process automation from unsegmented UI logs. *2nd International Conference on Process Mining, ICPM 2020, Padua, Italy, October 4-9, 2020*, 153–160.

Mayr, A., Herm, L., Wanner, J., & Janiesch, C. (2022). Applications and challenges of task mining: A literature review. *30th European Conference on Information Systems - New Horizons in Digitally United Societies, ECIS 2022, Timisoara, Romania, June 18-24, 2022.*

Petri, C. A., & Reisig, W. (2008). Petri net. *Scholarpedia*, *3*(4), 6477.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *CoRR*, *abs/2003.08271*.

Ribeiro, J., Carmona, J., Misir, M., & Sebag, M. (2014). A recommender system for process discovery. *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, *8659*, 67–83.

Rozinat, A., & van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior. *Inf. Syst.*, *33*(1), 64–95.

Schonenberg, H., Weber, B., Dongen, B. v., & Aalst, W. v. d. (2008). Supporting flexible processes through recommendations based on history. *International Conference on Business Process Management*, 51–66.

Seeliger, A., Nolle, T., & Mühlhäuser, M. (2018). Process explorer: An interactive visual recommendation system for process mining. *KDD Workshop on Interactive Data Exploration and Analytics.*

Sroka, M., & Fani Sani, M. (2022). 50 business assignments log. https://github.com/microsoft/50BusinessAssignmentsLog

Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J., Ouyang, C., ter Hofstede, A. H., van de Weerd, I., Wynn, M. T., & Reijers, H. A. (2020). Robotic process automation: Contemporary themes and challenges. *Computers in Industry*, *115*, 103162.

Terragni, A., & Hassani, M. (2018). Analyzing customer journey with process mining: From discovery to recommendations. *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 224–229.

van der Aalst, W. M. P. (2016). *Process Mining - Data Science in Action, Second Edition.* Springer Berlin Heidelberg.

van der Aalst, W. M. P., Rubin, V. A., Verbeek, H. M. W., van Dongen, B. F., Kindler, E., & Günther, C. W. (2010). Process mining: A two-step approach to balance between underfitting and overfitting. *Softw. Syst. Model.*, *9*(1), 87–111.

Wang, J., Wong, R. K., Ding, J., Guo, Q., & Wen, L. (2012). On recommendation of process mining algorithms. In C. A. Goble, P. P. Chen, & J. Zhang (Eds.), *2012 IEEE 19th international conference on web services, honolulu, hi, usa, june 24-29, 2012.* IEEE Computer Society.

Wanner, J., Hofmann, A., Fischer, M., Imgrund, F., Janiesch, C., & Geyer-Klingeberg, J. (2019). Process selection in RPA projects - towards a quantifiable method of decision making. *Proceedings of the 40th International Conference on Information Systems, ICIS, Munich, Germany, 2019.*