

# How Business Process Benchmarks Enable Organizations To Improve Performance

Ünal Aksu and Hajo A. Reijers  
 Department of Information and Computing Sciences  
 Utrecht University  
 Utrecht, The Netherlands  
 {u.aksu, h.a.reijers}@uu.nl

**Abstract**—The recurring but mutually distinct ways of executing a business process are referred to as process variants. There are approaches available in the literature aimed at finding such process variants and determining how they differ from each other. However, organizations are more interested in understanding the effect of these differences in terms of the performance of a business process. In this context, we propose a novel approach to enable organizations to learn from each other through business process benchmarks. To do so, the approach bins organizations based on what extent they achieve their performance targets in relation to their Key Performance Indicators (KPIs). Within each bin, process variants are identified using trace clustering. Then, significant differences among process variants are determined and highlighted. These differences help organizations to improve the performance of their processes. We implemented our approach, evaluated its performance, and applied it in a case study.

**Keywords**—process variants, trace clustering, benchmarking, key performance indicators, process mining

## I. INTRODUCTION

The way a business process is executed may differ across organizations, or in some branches of the same organization, or for certain products or services offered to customers, or even across organizational units. The term process variant is used for the subset of the executions of a business process that distinguishes from others due to various reasons. Identifying the similarities in and differences between process variants helps organizations to determine improvement opportunities and also actions to prevent undesired changes in the performance of the process. Furthermore, understanding the root causes of differences between process variants enables managers to make informed decisions for improving business processes. For example, one organizational unit may deliver very good performance by applying a best practice. An organization may want to disseminate that best practice across other units as well.

To determine and explain the similarities in and differences between two or more process variants, numerous

techniques are available in the literature. In most of these techniques, the event logs that are produced during the execution of a process and corresponding to two or more variants of that process are compared. For this comparison, trace clustering methods are primarily preferred since they are devised to determine homogeneous groups of process instances from the executions of a given process. To do so, process instances, i.e., traces, are grouped based on their similarities. Such similarity is computed either using the alignment between the sequences of traces [1]–[6] or from the feature vectors in each a trace is located and represented using its characteristics, e.g., frequency of activities [7]–[11]. However, these methods are complex and computationally expensive, especially when longer substrings are used for alignment. Similarly, feature vector-based methods have some limitations: either they only focus on the attributes of activities rather than the relations between activities or consider only proceed and succeed relations. Avoiding false positives while determining similar sequences in length-insensitive sequence problems is a challenge for some approaches, e.g., n-gram based. Another challenge for some other approaches is dealing with noise in the given sequences.

In this paper, we propose a novel approach to enable organizations to learn from each other by means of *business process benchmarks*. A business process benchmark shows how the same business process is performed among the organizations that are in the same context. The approach takes the process performance of organizations in terms of the KPIs that are relevant to them. Organizations that have similar goals in terms of their KPIs, i.e., interested in the same set of KPIs are grouped. Then, they are distributed into KPI bins based on what extent they achieve their process performance targets. The motivation for this is that organizations that have differences in their process performance are more likely to learn much from each other rather than from the organizations, which perform similarly with them.

Trace clustering is employed to identify the similarities in and differences between the process executions that yield better or worse performance. To address the aforementioned deficiencies of trace clustering methods, the

This work is a result of the AMUSE project. See [amuse-project.org](http://amuse-project.org) for more information.

approach adopts a sequence feature extraction technique (called SGT [12], [13]). By identifying and highlighting the significant differences between the obtained clusters in the previous step, the approach provides business process benchmarks. Organizations can use these benchmarks as the basis to identify both opportunities for improvement and actions to prevent undesired changes in the performance of their business processes. We implemented our approach and applied it in a real-life setting after evaluating its performance.

In Sect. 2, we provide the background on Sequence Graph Transform (SGT), which is the feature extraction technique that we employ in our approach for trace clustering. The approach is elaborated in Sect. 3. Implementation details of the approach are given in Sect. 4. Afterwards, in Sect. 5, we evaluate the performance of our approach, and then we apply it in a case study and discuss the obtained results in Sect. 6. In Sect. 7, we provide an overview of related work on trace clustering and organizational benchmarking. Finally, we present our conclusions and potential directions for future work in Sect. 8.

## II. THEORETICAL BACKGROUND

In the subsection below, we explain the feature extraction technique, Sequence Graph Transform (SGT), which we adopted from the area of data mining, and employ in our approach while determining trace clusters to create process benchmarks.

### A. Sequence Graph Transform (SGT)

Depending on the associations between events, a sequence can be either *feed-forward* or *undirected*. In the former case, in a forward direction, events follow one another. In the latter, the order of events does not depend on each other. In this paper, we focus on feed-forward sequences since the order of activities depends on one another in most business processes. In other words, the decisions that are made in a step determine the succeeding steps. For example, how an incident will be handled depends on all the decisions made regarding its categorization, prioritization, and severity determination that happen in the earlier steps of a typical incident management process.

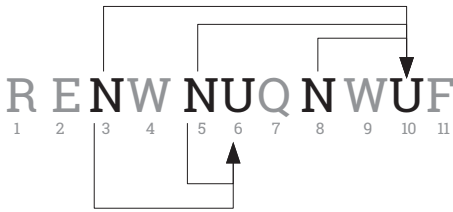


Figure 1: SGT: Showing “effects” on events on each other.

An example of a feed-forward sequence is shown in Fig. 1. The predecessors of *Event U* at positions 6 and 10

can be seen in the figure. The relative positions of one event at a time are taken to extract the sequence features. The relative positions for the event pair  $(N, U)$  are  $\{(3, 5), 6\}$  and  $\{(3, 5, 8), 10\}$ . From these positions sequence features will be extracted. Note that only “ $N$  leading to  $U$ ” can be interpreted using these positions. “ $U$  leading to  $N$ ” requires a different set of relative positions, i.e., where  $N$  is preceded by  $U$ .

Let  $\mathcal{V}$  denote a set of events in a given sequence. The associations between all events can be extracted to obtain sequence features in a  $|\mathcal{V}|^2$  dimensional feature space. On the one hand, the similarity in sequences can be measured based on these features. On the other hand, these features can be used to visualize the sequence as a directed graph. For the sake of simplicity, we only give the definitions for SGT below. We refer to the original study on SGT done by Ranjan et al. [12] in which the step by step derivation of these definitions and graph generation are explained.

Let  $\mathcal{S}$  denote a set of sequences,  $s \in \mathcal{S}$  denotes a sequence that consists of events in  $\mathcal{V}$ . A sequence can be built using one or more instances of an event from  $\mathcal{V}$ . For instance,  $\mathcal{S} = \{ABABCED, ABBE, ACEDD, \dots\}$  is built using the events in  $\mathcal{V} = \{A, B, C, D, E\}$ .  $\Lambda^s$  denotes the length of a sequence,  $s$  is equal to the number of events it contains. The event at position  $l$  in the sequence is denoted by  $s_l$ , where  $l = 1, \dots, \Lambda^s$  and  $s_l \in \mathcal{V}$ .

Let  $l$  and  $m$  be the positions of two events and  $d(l, m)$  a distance measure.  $\phi(d(l, m))$  denotes the quantification of the effect of the preceding event on the latter. Given that events  $u$  and  $v$  are at positions  $l$  and  $m$ ,  $\Psi_{uv}^s$  denotes the extraction of features of the sequence  $s$  in the form of associations between events, where  $u, v \in \mathcal{V}$  and  $\Psi$  is a function of an auxiliary function  $\phi$ . A distance  $d$  and a tuning hyper-parameter  $\kappa$  are taken as the inputs by the function  $\phi_\kappa(d)$ .

Based on the assumptions above, the derived feature extraction formulas both for length-sensitive (1) and insensitive sequence (2) analysis problems are taken from [12], [13] and listed below.

$$\Psi_{uv}(s) = \frac{\sum_{\forall(l, m) \in \Lambda_{uv}(s)} e^{-\kappa|m-l|}}{|\Lambda_{uv}(s)|} \quad (1)$$

$$= \frac{\sum_{\forall(l, m) \in \Lambda_{uv}(s)} e^{-\kappa|m-l|}}{|\Lambda_{uv}(s)|/L(s)} \quad (2)$$

where  $\Psi(s) = [\Psi_{uv}(s)]$ ,  $u, v \in \mathcal{V}$  is the SGT feature representation of sequence  $s$ .

The SGT feature for the event pair  $(N, U)$  in the sequence in Fig. 1 can be computed as (for  $\kappa = 1$  in length-sensitive

SGT):

$$\begin{aligned}\Lambda_{NU} &= \{(3, 6); (5, 6); (3, 10); (5, 10); (8, 10)\} \text{ and} \\ \Psi_{NU} &= \frac{\sum_{(l,m) \in \Lambda_{NU}} e^{-|m-l|}}{|\Lambda_{NU}|} \\ &= \frac{e^{-|6-3|} + e^{-|6-5|} + e^{-|10-3|} + e^{-|10-5|} + e^{-|10-8|}}{5} \\ &= 0.112\end{aligned}$$

As shown above, one can obtain each sequence's SGT features for a given set of sequences. Vector representations can be used for these features to determine similar sequences, i.e., clustering.

Since SGT is good at capturing short and long term dependencies in sequences, it enables avoiding from increasing computation when extracting long-term similarity patterns. Another advantage of SGT is related to the accurate comparison of sequences that have different lengths. Most subsequence matching techniques often lead false positives for the given sequences, sq1: *REN*, sq2: *RENWRENREN*, and sq3: *RENWNUNQ*. In particular, in these techniques, sq1 will be consider as similar to sq2 and sq3, which is due to the local alignment. To deal with, SGT considers mismatches inherently.

Moreover, SGT is robust to noise that is often observed in sequence problems. For the sake of simplicity, we will refer to the examples related to noise and discussed in [13]. Consider these sequences: a) *RENWFRE* and b) *RENWFRXE*. In b), a noise X appears between R and E. The Markov model transition probability for R and E in the two sequences will be as the following: 1.0 is for a) and 0.5 for b). However, SGT will handle the noise and calculate 0.5 for a) and 0.45 for b) when  $\kappa = 5$ . Thus, the effect of stochasticity can be managed.

Considering the aforementioned strengths of SGT, we use it in our approach for trace clustering. In the following section, we explain how we use SGT at the clustering task of our approach.

### III. APPROACH

In this section, we give the details of our approach for providing business process benchmarks to enable organizations to learn from each other. Our approach consists of four tasks: (1) selecting organizations and KPIs, (2) binning organizations, (3) clustering traces, and (4) benchmarking. Each task is detailed below in a separate subsection.

#### A. Selecting Organizations and KPIs

In order to determine which organizations may learn from each other, we need to identify whether they share the same context. To do so, we focus on Key Performance Indicators (KPIs) that organizations use to monitor whether they attain their performance goals. Our main motivation at

this point is that the KPIs that are relevant for organizations indicate that these organizations have similar goals with respect to their business processes. In one of our previous works [14], we showed that organizations that have a similar goal indeed have a shared interest in certain KPIs. As such, we select organizations with respect to the KPIs that are relevant to them. From a set of assessments of the relevance of certain KPIs for organizations, the approach determines for which organizations business process benchmarks can be created and based on which KPIs. The selected organizations and KPIs will be taken as inputs by the next task, i.e., binning organizations.

#### B. Binning Organizations

Organizations that have a difference in their process performance are more likely to learn much from each other rather than from the organizations, which perform similarly. In addition, the similarities in the business processes of the organizations that perform similarly can be interpreted as the reasons of their similar process performance. Therefore, in this task, the organizations that are selected in the previous task are distributed into KPI bins. These bins are determined using the KPIs, which are also selected in the previous task. More specifically, the organizations are grouped based on what extent they achieve their process performance target, which is set in their KPIs.

To determine KPI bins, target thresholds of KPIs are used as input in this task. Target thresholds are defined as value-range scales. They are used to interpret to what extent the target of a KPI is achieved [15]. Each threshold has a lower and upper bound value that are used for building the value-range set of a KPI. For example, good: [KPI target-10K, KPI target-30K], bad: [KPI target-30K, KPI target-50K]. Target thresholds of KPIs may vary from one organization to another. Therefore, in this task, KPI bins are determined from the perspective of the organization, which is benchmarking its process performance against others. Thus, our approach deals with the subjectivity of the target thresholds of KPIs.

Furthermore, the approach can handle the variation among the target thresholds of multiple KPIs. For instance, one KPI may have three target thresholds, whereas another may have five. In this situation, the approach creates a KPI bin for each threshold combination. We will indeed involve end-users to select the threshold combinations that are relevant to them. If there are KPIs that are affected by multiple processes, it is required to combine these multiple processes as a meta-process considering their inter-dependencies. Thus, the selected process discovery algorithms can deal with the event logs of multiple processes. Aside from that, case attributes of process models can be incorporated into binning organizations. However, a scaling strategy is necessary in that case to interpret the meaning of smaller or higher values regarding case attributes.

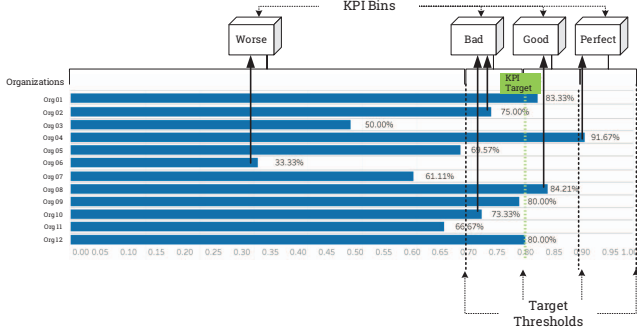


Figure 2: Binning organizations using KPI target thresholds

Fig. 2 depicts an example of binning organizations based on the target thresholds of a given KPI. In the figure, there are 4 KPI bins: *perfect*, *good*, *bad*, and *worse*. These bins are created based on the defined target thresholds for the given KPI. For example, for the bin *perfect*, 90 is the lower bound threshold value and 100 is the upper bound value, which is the maximum for this KPI. Similarly, the lower and upper bound values for each target threshold are shown in the figure. The organizations that will be put into each bin are depicted in the figure as well. For example, *Org 02* and *Org 10* belong to the bin *Bad*.

### C. Clustering Traces

In the third task of our approach, the focus is on identifying the similarities in and differences between the business process executions of the organizations in each KPI bin. These bins were determined in the task we described earlier. A sequence feature extraction technique (SGT, see in Sect. 2) is adopted and applied on the business process executions, i.e., event log of the organizations in each KPI bin. In particular, the event log for the process that is monitored by the previously selected KPIs is used while clustering.

An essential decision in clustering is the number of clusters to choose. In the literature, several approaches are available to propose the optimal number of clusters for a given clustering problem. However, the quality of the process models that will be discovered for the clusters from a given event log is not taken into consideration in these approaches [9]. Therefore, we opt for using a range for the number of clusters. In our approach, we rely on process owners to determine that range.

Traces are compared by locating each unique trace in a feature space using the *k-means* clustering algorithm. Due to its simplicity, we opted for this algorithm. SGT extracts the short and long term sequence features in the traces and embeds them in a finite-dimensional feature space. To facilitate the feature extraction, our approach shortens the activity names in the given event log. To do so, the approach employs a label encoder. Note that the approach does not convert activity names to numeric values: a string

prefix is combined with the numeric value created by the label encoder, e.g., *Assign* is shortened as *T0*.

Although SGT promises [12] no increase in the computation to reduce the size of the clustering problem, we remove duplicate traces in the given event log. Thus, the approach assigns equal weight to all unique traces. Assigning equal weights on traces enables our approach to handle the varying absolute frequency of a trace that may be seen in the event logs of multiple organizations. Aside from that, without any domain knowledge on a given process, one cannot easily assign a weight based on the frequencies of traces, which mostly varies per organization.

Since the curse of dimensionality is a known phenomenon when analyzing data that have a high-dimensional space, for this clustering task, a dimensionality reduction technique is applied on the generated feature space by SGT. Hence, the essential parts that have more variation of the data are preserved.

The gap between clustering and evaluation is the main challenge of clustering techniques as noted in [9]. In this context, there exist two ways to address this gap. The first way is evaluating the discovered process model for each cluster while clustering. However, this way is computationally expensive, especially for large event logs. Therefore, in our approach, the quality of the discovered process models are checked after clustering. This is done by using the common process mining metrics [16], namely replay fitness, precision, generalizability, and simplicity.

As a result of this task, we obtain the clusters for each KPI bin that best represent the variance in the behavior captured in a given event log. In the next and the last task of our approach, we focus on spotting significant differences between the discovered process models for these clusters.

### D. Benchmarking

Organizations can benefit from determining the differences between and the similarities in their business processes that yield better or worse performance. For instance, what yields a better performance in a business process can be interpreted as opportunities for improvement. Similarly, what yields a worse performance in a business process can become the basis for finding the actions to prevent any decline in the performance. In this context, the last task of our approach is devoted to identifying relevant differences between process models. They will be discovered from the event log of the clusters, which are created in the previous task.

In a recent survey on process variant analysis [17], approaches on determining differences between process executions are studied. We checked the applicability of these approaches for detecting statistically significant differences between process executions within our approach. Aligned with our goal, the approach proposed by Bolt

et al. came forward since it provides an extensible basis for process specific metrics in addition to KPIs. Moreover, that approach is available as a plugin (called Process Comparator) in the process mining framework, ProM [18], which offers built-in features for handling event logs. As such, in this task, we execute that plugin for the event logs of each cluster pair. The cluster pairs are formed by picking the best performing cluster of two KPI bins based on fitness, precision, and generalizability. For the balancing of these process model metrics, we used the order in which they are listed here. This forming step ends when all KPI bin combinations are checked. Then, the event logs for a selected cluster pair are analyzed by the plugin, and the statistical differences between these event logs are projected onto a transition system. In the transition system, states and transitions are colored to spot those differences. Moreover, the thickness of each node's borders and arcs in the same transition system indicate the frequencies of the states and transitions for each cluster in the selected cluster pair. In addition, a set of metrics for highlighting the differences from the control-flow (frequency) and time perspective (elapsed time, remaining time) provided by the plugin are used for the selected cluster pairs. Fig. 3 illustrates how the significant differences between the event logs of a selected cluster pair are highlighted. The nodes in the figure represent activities, whereas arcs reflect the sequence of these activities in a process. Thickness of the arcs and nodes are determined based on the value of the selected process metric. Similarly, opposite colors, i.e., red and blue, used to indicate significant differences between two event logs in terms of the selected process metric. For example, T5 is a significant activity in one cluster, whereas the activities T15 and T23 are significant in another cluster as the opposite colors are used for each (red vs. blue colors).

In the next section, we give the details of the implementation of the approach.

#### IV. IMPLEMENTATION

To demonstrate the feasibility of our approach and evaluate it in practice, we implemented it<sup>1</sup>. The implementation of the approach consists of two components. The first component covers the first three tasks of the approach (Selecting Organizations and KPIs, Binning Organizations, and Clustering Traces). This component is developed in Python as a script that can be directly executed. The second component implements the last task of the approach, i.e., benchmarking. It is developed in the form of a plugin of the ProM process mining framework [18].

In the first component, we reused the SGT implementation in Python that is publicly available<sup>2</sup>. In the reused

<sup>1</sup>The implementation of our approach is available at <http://amuse-project.org/software/>

<sup>2</sup>The SGT implementation in Python is available at <https://github.com/cran2367/sgt>

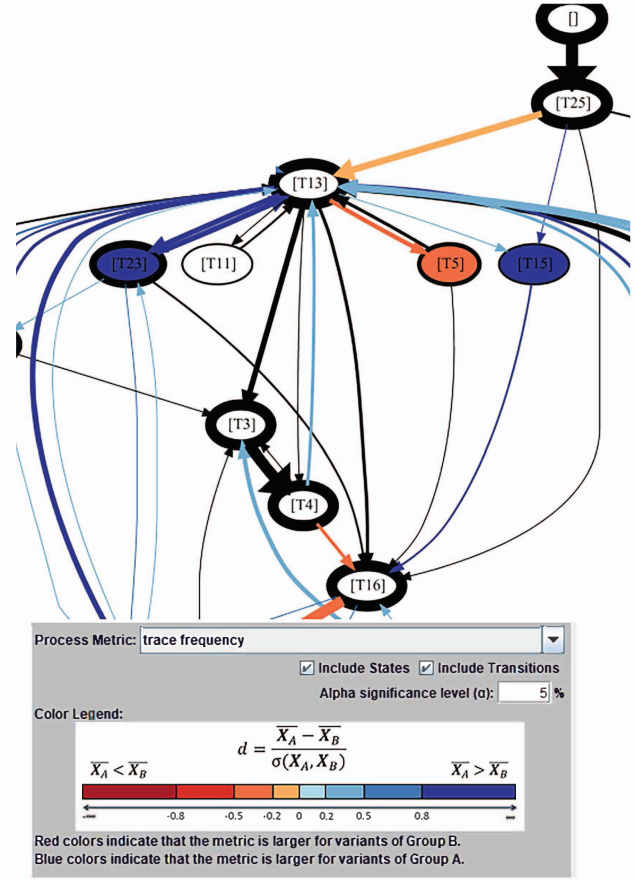


Figure 3: Detecting statistically significant differences between the event logs of a cluster pair for benchmarking

code, we configure two parameters that are relevant for our approach: *kappa* and *length-sensitivity*. The first parameter adjusts to what extent the long-term dependency will be captured while embedding the extracted sequence features in a finite-dimensional feature space. 5 is chosen as the value for this parameter to have a medium level dependency capture in the embedding. The second parameter is used to determine whether the length of sequences will be considered in embedding. The value of this parameter is set to *true* to capture the effects of trace lengths.

To deal with high dimensionality at feature extraction, we used the Principal Component Analysis (PCA) algorithm implementation in the *scikit-learn* Python machine learning library. We set the minimum variance coverage rate as 95% at dimensionality reduction since it is a commonly used value in similar problems.

As mentioned, the second component of the implementation is devoted to the benchmarking task. The component calls the Flexible Heuristics Miner (FHM) plugin [19] of the ProM framework with default parameters. The reason for using FHM is that it is mostly used in other trace clustering methods [1], [9], and recommended as a robust discovery

algorithm. The plugin reads the event log of each cluster and creates a heuristics net for each. Each heuristics net is converted to a petri net to replay [20] them on the log and compute fitness using the PNetReplayer plugin [21]. Similarly, the PNetAlignmentAnalysis plugin [22] is employed to compute precision and generalizability. Based on these computed metrics, our approach determines the clusters for each KPI bin that best represent the variance in the behavior captured in a given event log.

In the next section, we evaluate the performance of the approach by comparing it with state-of-the-art trace clustering approaches. Then, in Sect. 6, we apply our approach in a case study.

## V. PERFORMANCE EVALUATION

In this section, we explain how we evaluate the performance of our approach. Specifically, we compare the performance of the trace clustering task of our approach with state-of-the-art trace clustering methods. This is relevant because this task adopts an existing clustering method. For the performance evaluation, we followed the same scenario (Evaluation Scenario 2) described by Evermann et al. in the study [1] as it serves the same purpose: the method proposed in that study is compared with state-of-the-art trace clustering methods. The steps of that scenario are listed below.

- 1) Identify a set of configurations for the adopted trace clustering method based on the cluster range used in state-of-the-art trace clustering methods:
- 2) For each identified configuration:
  - a) Apply the configuration to the adopted trace clustering method
  - b) Perform clustering on the given event log
  - c) *Apply Flexible Heuristic Miner (FHM)* to discover a heuristics net
  - d) Convert the heuristics net to a Petri net
  - e) Compute performance metrics: simplicity (CN, CNC, and delta)
  - f) Replay the event log on the Petri net using *PNetReplayer*
  - g) Compute performance metrics: fitness
  - h) Replay the event log on the Petri net using *PNetAlignmentAnalysis*
  - i) Compute performance metrics: precision and generalizability
- 3) Compare the performance of the adopted trace clustering method with the performance of state-of-the-art trace clustering methods

We use the same set of performance metrics as well as the event log, which is taken from a loan application process and publicly available<sup>3</sup>. We compare the obtained

Table I: Performance of state-of-the-art trace clustering methods vs. our approach

Conf	CNC	CN	Delta	Fit	Prec	Gen
AT-3	1.1670	33.7120	0.0240	0.7000	0.4737	0.7322
AT-6	1.1198	26.196	0.0326	0.6670	0.5663	0.6011
AT-6-ICS95	1.1709	27.3087	0.0072	0.8529	0.3751	0.9661
DWS-Std	1.2275	30.8013	0.0103	0.8783	0.3219	0.9586
DWS-55510	1.1579	17.396	0.0208	0.7721	0.5459	0.9581
TC	1.1773	31.6533	0.0270	0.7823	0.4062	0.7419
TC-W1-H3	1.2434	46.1867	0.0129	0.8213	0.3232	0.8937
TC-W2-H3	1.1792	35.1787	0.0217	0.7235	0.4413	0.8037
TC-W3-H3	1.1542	31.6587	0.0279	0.6991	0.4686	0.7309
TC-W4-H3	1.1542	31.6587	0.0279	0.6991	0.4840	0.7332
SC-3	1.1976	32.9653	0.0071	0.8475	0.3631	0.9598
SC-6	1.1346	20.5973	0.0071	0.8644	0.5103	0.9905
SC-9	1.1273	17.7667	0.0078	0.8623	0.5408	0.9335
SC-12	1.1048	13.6540	0.0083	0.8607	0.5502	0.9628
SC-15	1.0974	12.0793	0.0104	0.8919	0.5760	0.9793
AC-maxPrec [mmP=-0.5, cGO=0.5, cGE=1, useSim=F, dim=log, c=9]	1.1201	15	0.0094	0.8036	0.5992	0.9957
AC-maxGen [mmP=-0.5, cGO=0.5, cGE=0.5, useSim=F, dim=log, c=3]	1.1507	22	0.0073	0.7775	0.5158	0.9988
AC-maxFit [mmP=-0.5, cGO=0.5, cGE=0.5, useSim=F, dim=log, c=9]	1.1209	14	0.0104	0.854	0.5679	0.9965
AC-minCNC [mmP=-1, cGO=1, cGE=1, useSim=F, dim=log, c=9]	1.1164	13	0.0108	0.8502	0.5872	0.9969
AC-minCN [mmP=-1, cGO=1, cGE=1, useSim=F, dim=log, c=9]	1.1164	13	0.0108	0.8502	0.5872	0.9969
AC-minDelta [mmP=-0.5, cGO=0.5, cGE=0.5, useSim=F, dim=sqrt, c=3]	1.1840	30	0.0072	0.7888	0.4767	0.9941
Our Approach [SGT_kappa=5, SGT_length-sensitive=T, c=3]	1.1765	20	0.0100	0.9823	0.3339	0.9860
Our Approach [SGT_kappa=5, SGT_length-sensitive=T, c=6]	1.1220	19	0.0069	0.9678	0.5213	0.9985
Our Approach [SGT_kappa=5, SGT_length-sensitive=T, c=9]	1.0909	11	0.0083	0.9837	0.6292	0.9968

results with the performance measurements for state-of-the-art trace clustering methods, which are available in the aforementioned study. In Table I, the abbreviations used in the column *Conf* refer to the following clustering methods respectively, *Active Trace Clustering* (ActiTraC) [9], *Disjunctive Workflow Schema Mining* [11], *Trace Clustering* [8], *Sequence Clustering* [5], and *AlignCluster* [1].

In these three configurations, we used several values for the number of clusters. The values that we used as the number of clusters are taken from the aforementioned study. Specifically, we used 3, 6, and 9 as the number of clusters. The results that we obtained are shown in Table I.

<sup>3</sup>The event log is available at <https://joerg.evermann.ca/software.html>

In the table, the best values for each quality criterion are shaded; the worst values are italicized.

As can be seen in the table, 4 out of the 6 criteria (CNC, CN, Fit, Prec), the last configuration that we used in our trace clustering method outperformed state-of-the-art trace clustering methods. In particular, our approach provides simpler process models with better precision and fitness. Moreover, for the remaining 2 criteria (Delta and Gen), in that configuration, we obtained values that are very close to the best value. However, the first configuration that we used has a lower precision, but very good fitness and generalizability. In addition, in this configuration, our approach provides somewhat simpler process models. Although we used a limited set of configurations, this evaluation shows that our approach generally achieves a better performance than state-of-the-art trace clustering methods.

In the following section, in a case study, we apply the technique in a real-life setting.

## VI. CASE STUDY

In this section, first, we introduce the case study organization where we applied and evaluated our approach for creating process benchmarks. Second, we explain how we collect the data that are used in the evaluation. Third, we list the experts who interpreted the results that we obtained in the application of our approach. Fourth, we elaborate on the application of our approach to the collected data. Finally, we discuss the relevance of the results to the case study organization.

### A. Case Study Organization

One of the biggest educational institutions in the Netherlands expressed its interest to us in learning from the data recorded about its IT Service Management (ITSM) processes. One of the vital processes among them is the Incident Management (IM) process. That process defines the way of managing questions, requests, and, most importantly, malfunctions about the products and services offered by the institution. Moreover, the process is executed through a third-party software. Since the institution offers a wide range of products and services to more than 25K customers (20K students, 5K employees), multiple organizational units are involved in this IM process. Each organizational unit is devoted to ensuring the quality of a particular set of offered products and services. For example, cloud based email and printing are two services that are managed by separate organizational units.

To monitor customer satisfaction and provide the same level of quality for each product and service, the institution has a set of KPIs. These KPIs have the same target for every organizational unit that is involved in the same ITSM processes. Specifically, there are some KPIs for the IM process, and all of them are used to determine whether

each organizational unit attains the shared goal of the IM process. Performance is tracked by monthly reports. In 2019, a few of the organizational units were not able to achieve their objectives about dealing with malfunctioning of a number of products and services. Therefore, the institution wanted to investigate what the differences and similarities are in the execution of the IM process. Moreover, the managers involved in the IM process assumed that some organizational units may follow best practices, which would explain why they would provide a better performance.

Since both the setting and needs in the educational institution are highly related to the approach that we propose in this paper, its application in this context is highly relevant to determine its applicability and value.

### B. Data Collection

In accordance with the problem that the case study organization faced in 2019, we extracted the event log for the malfunctions processed in that year. Specifically, the event log consists of 150K events and 12K cases in which 26 unique activity and 40 unique organizational units exist. The time frame of the event log is one year. In addition to the required minimum attributes, the event log contains to what extent the resolution time of a malfunction adheres to the defined target of the single KPI for malfunctions. The KPI has a defined target value, which is 80% and there are four target thresholds: [0 – 69] *worst*, [70 – 79] *borderline*, [80 – 89] *sufficient*, and [90 – 100] *best*.

Before applying our approach to the collected data, together with the experts listed below, we checked the number of cases per organizational unit. At this check, we found out some outliers. In other words, very small number of malfunctions are received and handled by some organizational units that are specialized on occasionally used services and products. The experts suggested to filter out the organizational units that handled less than 50 malfunctions in a year to eliminate the potential bias that may be caused by these infrequent malfunctions. This filtering out cannot be considered as an issue for the applicability of our approach since more than 75% of the filtered out organizations received less than 10 malfunctions in a year. Moreover, such a low of number malfunctions is not adequate to determine relevant patterns in the process executions from which other organizational units can benefit and learn much. As a result of the filtering, in total, 24 organizational units remained for our analysis purposes.

### C. Applying the approach

We applied our approach using the collected data in the case study organization. As explained in the approach, in the first task, the organizations that may learn from each other will be selected based on the KPIs that are

Table II: Experts involvement in the evaluation of the obtained results

Expert	Area of expertise	Years of expertise	Meeting duration (hours)
Process manager	Incident management	> 10	3
Process manager-2	Incident management and Change management	> 10	2
First-line support manager	Incident management	> 20	2.5
Product and service manager	Educational services	> 15	1.5

relevant them. Accordingly, we selected the remained 24 organizational units since there is only one particular KPI used for monitoring the IM process for malfunctions.

In the second task, our approach created 4 KPI bins based on the given 4 target thresholds for the given KPI. These are: KPI bin-worst, KPI bin-borderline, KPI bin-sufficient, and KPI bin-best. The selected 24 organizational units are distributed to these KPI bins based on what extent they achieved their process performance. The number of organizational units put into the created each bin is 12, 6, 5, and 1, respectively.

In the third task, trace clustering was performed. As the number of clusters, together with the experts in Table II, we determined the value range from 3 to 9. Considering this input, for each KPI bin, clusters are created. Afterwards, for each KPI bin-cluster, a process model is discovered and replayed on the corresponding event log to compute the three process mining metrics, namely precision, fitness, and generalizability.

Finally, in the last task, the clusters that best capture the observed behavior in each KPI-bin are selected. This is done using the process mining metrics mentioned above. Thus, we obtained all the cluster pairs that are relevant for checking the significant differences between them. Accordingly, we imported the event logs of each cluster pair separately into the Process Comparator plugin, and then obtained the highlighted transition system that is the projection of the differences between the event logs as the final results (i.e., business process benchmarks) of our approach. We interviewed with experts and asked each one to explain whether they see any performance improvement opportunities. As a result of their combined feedback, in 4 out of the 6 business process benchmarks, we observed relevant points that can enable the organizational units in the case study organization to learn from each other for improving the performance. These business process benchmarks are depicted in the fragments of Fig. 4. In the remainder of this section, we discuss of these.

#### D. Discussion

Together with the experts (see Table II) in the case study organization, we analyzed the differences that are highlighted in the selected business process benchmarks. The business process benchmarks are determined by the experts as the ones that have the most potential to learn from for improving the IM process. In the first three figures below, red colors are used to indicate the greater value in the frequency of the activity in the KPI bin-cluster that has a lower process performance achievement than the paired one with a bigger cluster value. Similarly, blue colors are used to indicate the greater value in the frequency of the activity in the KPI bin cluster that has a better process performance achievement than the paired one with a smaller cluster value. In the last figure, the colors serve similar purposes for the KPI bins but indicate duration values.

Fig. 4a depicts the highlighted statistically significant differences between the process execution of the organizational units (KPI bin-worst and KPI bin-borderline) that could not attain their KPI goals for the IM process. In the IM process, T0 and T1 are the activities that are related to hand-overs, whereas T22 and T23 correspond to interactions with callers. As shown in the figure, while T0 and T1 have red colors, T22 and T23 have blue colors. The experts interpreted this difference as the following: the organizational units in the KPI bin-borderline tend to add more information to the received malfunctions in terms of comments or questions. Then, they stop the SLA timer by moving malfunctions to the callers. However, the organizational units in the KPI bin-worst do hand-over the malfunctions either to other operators or organizational units. Therefore, the time spent in handovers increase and yield lower process performance. The experts suggest that these organizational units should use the comments area so that other operators will get notified and involved in the malfunction.

Fig. 4b shows the differences between the organizational units that were close to achieving the process performance and the organizational units that have already achieved it. With this, we aimed to find out what may be the quick wins to improve the process performance of the organizational units in the KPI bin-borderline. As depicted, T2 and T5 are the two red-colored activities. In the IM process, while the activity T2 is related to supplier involvement, T5 is about send backs between the second-line support and the first-line support. The significant frequency of these activities grabbed the interest of experts. The experts interpret this situation as a ping-pong behavior for the malfunctions in which suppliers are involved. Moreover, this is mentioned as an unexpected situation since several big suppliers are already integrated into the IM software to avoid such send backs. To deal with this situation and identify new supplier

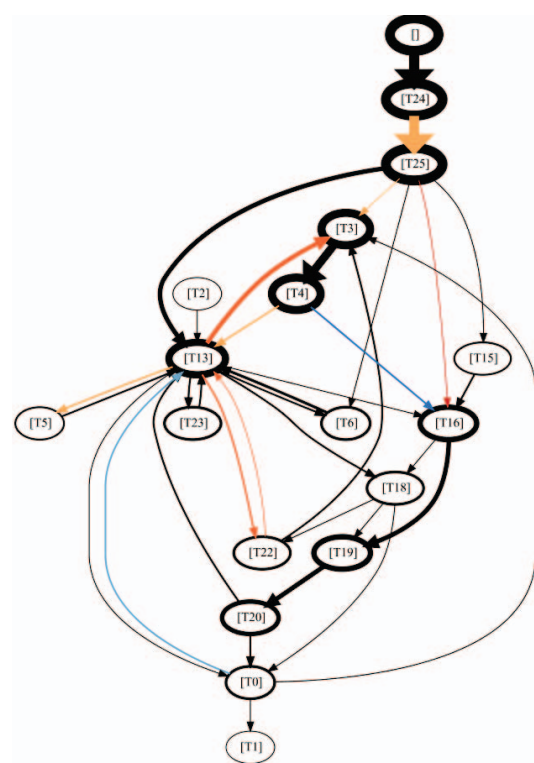
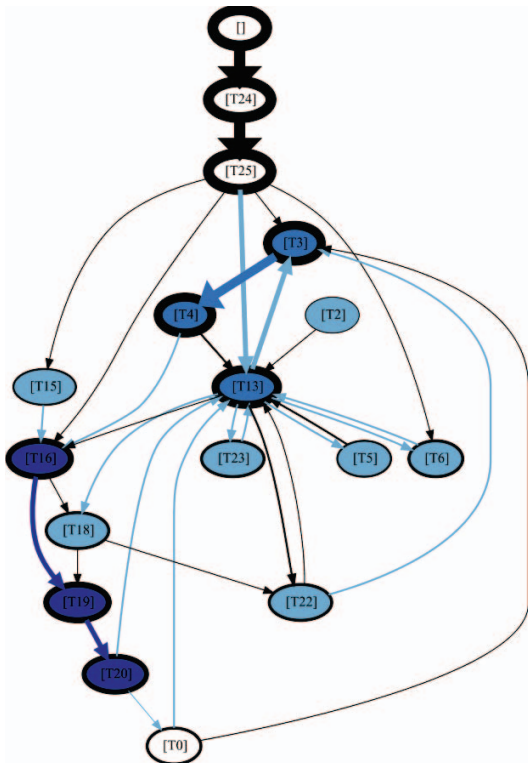
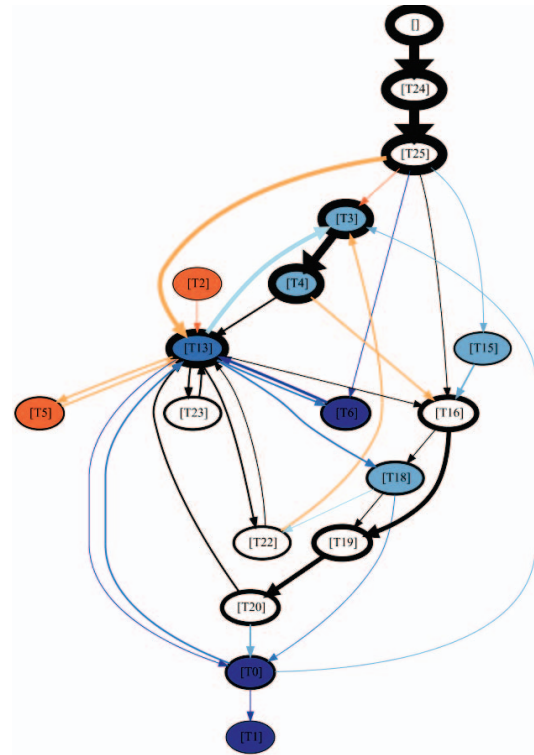
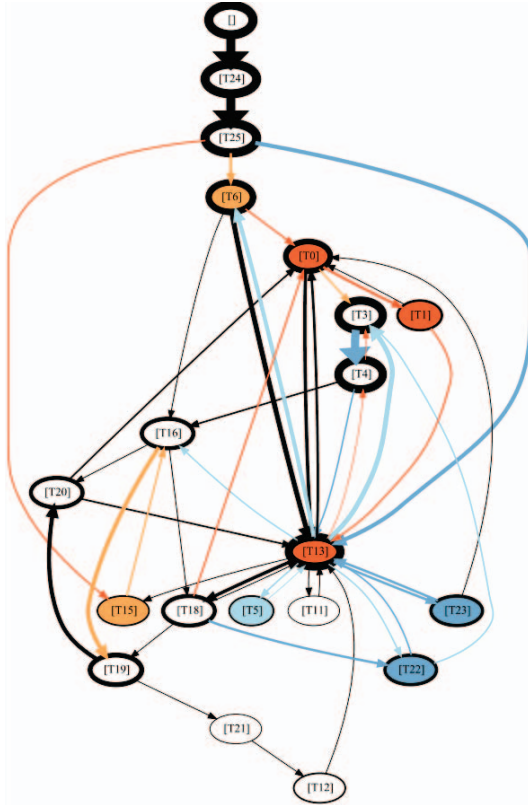


Figure 4: Business process benchmarks

integration needs, together with experts, we carry out the cluster analysis of the malfunctions in which suppliers are involved.

The differences between the organizational units that have attained their KPI goals are shown in Fig. 4c. As can be seen in the figure, the light-blue colored activities happen less frequently in the best performing organizational units. The experts mentioned that T23 is a specific activity that is about the confirmation from callers for resolved malfunctions. We checked that there is only one organizational unit in the KPI bin-best. This organizational unit has a face-to-face interface with callers. Therefore, the organizational unit gets confirmation from callers, meanwhile resolving malfunctions. Thus, the confirmation activity is mostly not necessary for this organizational unit. Moreover, the experts mentioned that although there is a confirmation collection step in the process, this activity is not frequent in the discovered process models as expected. Currently, the experts redesign the IM process to remove the necessity of a caller confirmation for resolved malfunctions in certain situations.

In Fig. 4d, a time perspective (i.e., duration) is chosen: the differences between the organizational units that were close to achieving the process performance and the organizational units that have already achieved it are highlighted. The experts indicated two important points based on the figure. Firstly, the activity T5 shows the delay due to the send backs between the first-line support and the second-line support. Secondly, the organizational units that perform sufficiently spend more time (from T0 to T13) on the discussions for determining operators for resolving malfunctions. To find the reason behind the second point, together with experts, we analyzed the malfunctions of the KPI bin-sufficient organizational units. In particular, we checked the organizational units that are involved in commenting on malfunctions for determining resolving operators. From these, we observed that some of the organizational units prefer face-to-face discussion if the involved operators are in their neighbor organizational units. The experts interpret this as a strategy to avoid send backs and potential delays. Based on this observation, together with experts, we investigate the correlation between non-neighbor organizational unit involvements and process performance achievement.

Based on the discussions and the results, the case study clearly confirms that the approach proposed in this paper can enable different parts of an organization to learn from each other. Moreover, software vendors that offer software products to their client organizations may benefit by applying our approach. In particular, those clients can be benchmarked based on their process performances since they use the same software product offered by the same vendor. In addition, governmental bodies can learn much from each other through process benchmarks since the

majority of the processes they perform often overlap.

Domain expert involvement is a key element in identifying improvement opportunities and prevention actions from process benchmarks. While clustering traces, equal weights are assigned on traces to capture all behaviors in the events logs of all organizations. Assigning varying weights on traces may reduce the dependency on domain experts. For example, building a drill-down mechanism on process benchmarks and focusing on mainstream behavior may help non-business users to interpret the significant differences between the mainstream behavior of organizations.

It is important to note that the labels and meanings of the activities in a business process, which is performed in several organizations may vary. This is an important challenge in addition to gathering and pre-processing the data of such process from different information systems in those organizations. In our previous works [23], [24], we discussed such challenges by introducing the approaches, which we proposed to meet them. The proposed approach in this paper concludes our framework introduced in [23]. Therefore, our approach relies on the internal outputs of our framework that is aimed at meeting the aforementioned challenges.

## VII. RELATED WORK

In this section, we list some of the works that relate to the approach we proposed for enabling organizations to learn from each other using process benchmarks. The most recent survey [17] on process variant analysis served as the basis for the snowball search method that we used for finding the works.

An analytical approach proposed by Buijs and Reijers [25] compares the variants of a process based on the alignments between the executions and the model of that process, which is performed by multiple governmental bodies. Partington et al. [26] compared patient pathways across four hospitals to visually detect the differences in process variants from which the hospitals can learn from each other. Trace clustering is combined with data and text mining in [27] to find the patterns that drive certain control-flow behavior in process variants.

Furthermore, some works [28]–[31] focused on advancing visualizations to ease the identification of differences between process variants.

Ballambettu et al. [32] annotated the process models of process variants with a set of metrics to detect the key differences between these variants. Nguyen et al. [33] introduced Perspective Graphs that are graph abstractions of event logs using a set of relations between entities (e.g., resource, location, etc.) in the event logs. These perspective graphs are then employed to compare process variants and identify the differences between them. Similarly, event logs for process variants are projected onto transition systems

by Bolt et al. in [34]. After that, these transition systems are compared to detect the significant differences in their states and transitions in terms of commonly used metrics (i.e., frequency, duration, etc.). However, in these works, the main focus mostly is on where process variants differ and how. Which differences matter is only determined based on a limited set of metrics rather than relevant KPIs for organizations. Moreover, domain experts need to go through each process variant pair. The first two tasks in our approach are aimed to cope with these challenges.

### VIII. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel approach aimed at enabling organizations or organizational units to learn from each other through business process benchmarks. As the organizations that are in the same context will learn much from each other, for benchmarking, we used the KPIs that are relevant for them.

To identify which variations in the process executions among organizations yield a better or worse performance, trace clustering is employed in our approach. As the order of tasks mostly depend on one another in business processes, a sequence feature extraction technique proposed to address this challenge is used within the clustering traces task of the approach. Then, significant differences between the process variants in the obtained clusters are determined as benchmarks.

The evaluation of our approach consists of two steps. First, we tested the performance of the trace clustering method in our approach by comparing it with state-of-the-art trace clustering methods. The result of this test showed that our trace clustering method has a better performance than other methods. Secondly, we applied our approach in a case study in an educational institution. We discussed the results that we obtained in that case study. This discussion showed that our approach is very helpful to reveal useful insights for organizational units to learn from each other. As the proposed approach is generic in the way it is developed, there is no limitation for applying it across organizations. By doing so, organizations can learn from each other to improve their performance.

In future work, we want to extend our approach by making it goal-driven. For example, an organization may be interested in learning from other organizations to identify only preventive actions to eliminate the risk of performing badly. As such, that organization will want to observe what similarities are present in the process variants of inefficient organizations. Moreover, we would like to incorporate decision mining technologies in our approach since the decision points in a business process are very likely to be associated with its process variants. Lastly, the approach can be extended to capture dependencies between business processes in organizations to provide cross-benchmarks for organizations.

### REFERENCES

- [1] J. Evermann, T. Thaler, and P. Fettke, "Clustering traces using sequence alignment," in *Business Process Management Workshops - BPM, 13th International Workshops, Innsbruck, Austria. Revised Papers*, 2015, pp. 179–190.
- [2] D. R. Ferreira, M. Zacarias, M. Malheiros, and P. Ferreira, "Approaching process mining with sequence clustering: Experiments and findings," in *Business Process Management, 5th International Conference, BPM, Brisbane, Australia. Proceedings*, 2007, pp. 360–374.
- [3] B. Hompes, J. Buijs, W. M. P. van der Aalst, P. Dixit, and J. Buurman, "Discovering deviating cases and process variants using trace clustering," in *27th Benelux Conference on Artificial Intelligence, Hasselt, Belgium. Proceedings*, 2015.
- [4] R. P. J. C. Bose and W. M. P. van der Aalst, "Context aware trace clustering: Towards improving process mining results," in *Proceedings of the SIAM International Conference on Data Mining, SDM, Sparks, Nevada, USA*, 2009, pp. 401–412.
- [5] G. M. Veiga and D. R. Ferreira, "Understanding spaghetti models with sequence clustering for prom," in *Business Process Management Workshops, BPM International Workshops, Ulm, Germany. Revised Papers*, 2009, pp. 92–103.
- [6] X. Lu, S. A. Tabatabaei, M. Hoogendoorn, and H. A. Reijers, "Trace clustering on very large event data in healthcare using frequent sequence patterns," in *Business Process Management - 17th International Conference, BPM, Vienna, Austria. Proceedings*, 2019, pp. 198–215.
- [7] S. Jablonski, M. Röglinger, S. Schöning, and K. M. Wyrski, "Multi-perspective clustering of process execution traces," *Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model.*, vol. 14, pp. 2:1–2:22, 2018.
- [8] M. Song, C. W. Günther, and W. M. P. van der Aalst, "Trace clustering in process mining," in *Business Process Management Workshops, BPM International Workshops, Milano, Italy. Revised Papers*, 2008, pp. 109–120.
- [9] J. D. Weerd, S. K. L. M. vanden Broucke, J. Vanthienen, and B. Baesens, "Active trace clustering for improved process discovery," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2708–2720, 2013.
- [10] G. Greco, A. Guzzo, L. Pontieri, and D. Saccà, "Discovering expressive process models by clustering log traces," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1010–1027, 2006.
- [11] A. K. A. de Medeiros, A. Guzzo, G. Greco, W. M. P. van der Aalst, A. J. M. M. Weijters, B. F. van Dongen, and D. Saccà, "Process mining based on clustering: A quest for precision," in *Business Process Management Workshops, BPM International Workshops, BPI, BPD, CBP, ProHealth, RefMod, semantics4ws, Brisbane, Australia. Revised Selected Papers*, 2007, pp. 17–29.
- [12] C. Ranjan, S. Ebrahimi, and K. Paynabar, "Sequence graph transform (SGT): A feature extraction function for sequence data mining," *CoRR*, vol. abs/1608.03533, 2016.

- [13] —, “Sequence graph transform (sgt): A feature embedding function for sequence data mining (extended version),” arXiv, 2016.
- [14] Ü. Aksu, D. M. M. Schunselaar, and H. A. Reijers, “Automated prediction of relevant key performance indicators for organizations,” in *Business Information Systems - 22nd International Conference, BIS, Seville, Spain. Proceedings*, 2019, pp. 283–299.
- [15] Ü. Aksu, A. del-Río-Ortega, M. Resinas, and H. A. Reijers, “An approach for the automated generation of engaging dashboards,” in *On the Move to Meaningful Internet Systems: OTM Conferences - Confederated International Conferences: CoopIS, ODBASE, C&TC, Rhodes, Greece. Proceedings*, 2019, pp. 363–384.
- [16] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.
- [17] F. Taymouri, M. L. Rosa, M. Dumas, and F. M. Maggi, “Business process variant analysis: Survey and classification,” *CoRR*, vol. abs/1911.07582, 2019.
- [18] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, “The prom framework: A new era in process mining tool support,” in *Applications and Theory of Petri Nets, 26th International Conference, ICATPN, Miami, USA. Proceedings*, 2005, pp. 444–454.
- [19] A. J. M. M. Weijters and J. T. S. Ribeiro, “Flexible heuristics miner (FHM),” in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM, Paris, France*, 2011, pp. 310–317.
- [20] A. Rozinat and W. M. P. van der Aalst, “Conformance checking of processes based on monitoring real behavior,” *Inf. Syst.*, vol. 33, no. 1, pp. 64–95, 2008.
- [21] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen, “Replaying history on process models for conformance checking and performance analysis,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 182–192, 2012.
- [22] A. Adriansyah, J. Munoz-Gama, J. Carmona, B. F. van Dongen, and W. M. P. van der Aalst, “Alignment based precision checking,” in *Business Process Management Workshops - BPM International Workshops, Tallinn, Estonia. Revised Papers*, 2012, pp. 137–149.
- [23] Ü. Aksu, D. M. M. Schunselaar, and H. A. Reijers, “A cross-organizational process mining framework for obtaining insights from software products: Accurate comparison challenges,” in *18th IEEE Conference on Business Informatics, CBI, 29th August - 1st September, Paris, France, Volume 1 - Conference Papers*, 2016.
- [24] —, “An approach for automatically deriving key performance indicators from ontological enterprise models,” in *Proceedings of the 7th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA), Neuchâtel, Switzerland, December 6-8, 2017*.
- [25] J. C. A. M. Buijs and H. A. Reijers, “Comparing business process variants using models and event logs,” in *Enterprise, Business-Process and Information Systems Modeling - 15th International Conference, BPMDS, Thessaloniki, Greece. Proceedings*, 2014, pp. 154–168.
- [26] A. Partington, M. T. Wynn, S. Suriadi, C. Ouyang, and J. Karnon, “Process mining for clinical processes: A comparative analysis of four australian hospitals,” *ACM Trans. Management Inf. Syst.*, vol. 5, no. 4, pp. 19:1–19:18, 2015.
- [27] J. D. Weerdt, S. K. L. M. vanden Broucke, J. Vanthienen, and B. Baesens, “Leveraging process discovery with trace clustering and text mining for intelligent analysis of incident management processes,” in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC, Brisbane, Australia*, 2012, pp. 1–8.
- [28] C. Cordes, T. Vogelgesang, and H. Appelrath, “A generic approach for calculating and visualizing differences between process models in multidimensional process mining,” in *Business Process Management Workshops - BPM International Workshops, Eindhoven, The Netherlands. Revised Papers*, 2014, pp. 383–394.
- [29] M. T. Wynn, E. Poppe, J. Xu, A. H. M. ter Hofstede, R. Brown, A. Pini, and W. M. P. van der Aalst, “Processprofiler3d: A visualisation framework for log-based process performance comparison,” *Decis. Support Syst.*, vol. 100, pp. 93–108, 2017.
- [30] J. Gulden, “Visually comparing process dynamics with rhythm-eye views,” in *Business Process Management Workshops - BPM International Workshops, Rio de Janeiro, Brazil. Revised Papers*, 2016, pp. 474–485.
- [31] R. Andrews, S. Suriadi, M. T. Wynn, A. H. ter Hofstede, A. Pika, H. H. Nguyen, and M. La Rosa, “Comparing static and dynamic aspects of patient flows via process model visualisations,” *Information and Software Technology*, 2016.
- [32] N. P. Ballambettu, M. A. Suresh, and R. P. J. C. Bose, “Analyzing process variants to understand differences in key performance indices,” in *Advanced Information Systems Engineering - 29th International Conference, CAiSE, Essen, Germany. Proceedings*, 2017, pp. 298–313.
- [33] H. Nguyen, M. Dumas, M. L. Rosa, and A. H. M. ter Hofstede, “Multi-perspective comparison of business process variants based on event logs,” in *Conceptual Modeling - 37th International Conference, ER, Xi’an, China. Proceedings*, 2018, pp. 449–459.
- [34] A. Bolt, M. de Leoni, and W. M. P. van der Aalst, “Process variant comparison: Using event logs to detect differences in behavior and business rules,” *Inf. Syst.*, vol. 74, pp. 53–66, 2018.