# Preprocessing Event Data in Process Mining

Mohammadreza Fani Sani[1][0000−0003−3152−2103]

Process and Data Science Chair, RWTH Aachen University, Aachen, Germany
fanisani@pads.rwth-aachen.de

**Abstract.** Process mining aims at obtaining insights into business processes by analyzing event data recorded in information systems. Many of current process mining algorithms have difficulties dealing with real event data because of two main reasons. Sometimes noisy and infrequent behavior in event data leads to having complex and incomprehensible results. Furthermore, some process mining algorithms are inapplicable for large event data using normal hardware. In this research, we aim to provide some general preprocessing approaches to deal with the above problems. Using these approaches, we aim to decrease the size and complexity of event data and consequently improve the performance of many process mining algorithms when holding similar results. Some of these approaches are also able to improve the quality of some process mining results. We also discuss some of the challenges of this research.

**Keywords:** Process Mining · Preprocessing · Event Log Preprocessing · Quality Improvement · Performance Improvement.

CEUR-WS.org/Vol-2613/paper1.pdf

## 1 Introduction

*Process Mining* aims to bridge the gap between traditional data mining and business process management analysis [1]. In this field of study, we extract knowledge from event data, also referred to as *event logs*, readily available in most current information systems. The main three sub-fields of process mining are 1) *process discovery*, i.e, finding a descriptive model of the underlying process, 2) *conformance checking*, i.e, monitoring and inspecting whether the execution of a process in reality conforms to the corresponding designed (or discovered) reference process model, and 3) *enhancement*, i.e, the improvement of a process model, based on related event data [1]. In all of the mentioned sub-fields, event logs are used as a starting point. An event log is a collection of events extracted in the context of a process that indicates which activity has happened at a specific time.

Many of the proposed algorithms in each of the above categories are perfectly performing on synthetic event data; however, they are not useful or applicable for real scenarios because of two main reasons. First, real event data usually contains noise and/or infrequent behavior. Therefore, the statement "garbage in, garbage out", i.e., referring to the fact that low-quality data leads to low final quality knowledge [22] also applies to the field of process mining. For example, various automated process discovery algorithms work perfectly on synthetic

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

event data; but, dealing with real event data they usually return complex and incomprehensible process models concealing the correct and/or relevant behavior of the underlying process. Using event log preprocessing, we are able to improve the quality of process mining results [7]. Second, by increasing the size and variability of event logs in different information systems, many of process mining algorithms, e.g., conformance checking and trace clustering, are no longer feasible using standard hardware in limited time. Therefore, similar to the general data mining domain, we require some preprocessing steps to obtain event data which leads to having process mining results faster.

In this research, we have the following research questions.

- Can we have process mining results with higher quality by preprocessing event logs?
- Is it possible to improve the performance of process mining algorithms using the preprocessed event logs?
- Do process mining results on preprocessed event logs are comparable with the cases that we used original event logs?

Therefore, we aim to provide some general preprocessing approaches that help a wide range of process mining algorithms. Using these approaches, we are able to apply the currently developed process mining algorithms directly on the preprocessed event data and it is not necessary to modify these algorithms. Moreover, we expect that the preprocessing algorithms require as little as possible business knowledge from the end-user. For example, to remove outlier behavior in event data, we assume that the user does not know the underlying process.

Note that, some process mining algorithms, e.g., conformance checking, are able to provide the accurate results. So, it is expected that by using preprocessing algorithms we can improve just their performances.

The remainder of this paper is structured as follows. Section 2 provides some related work in the area of preprocessing in process mining. The research approach is explained in Section 3. In Section 4, we show the results of applying some of the preprocessing methods. Thereafter, a couple of challenges that we have in this research are given in Section 5. Finally, Section 6 concludes the paper.

## 2 Related Work

A plethora of different process mining techniques exists, ranging from process discovery to prediction. However, given the focus of this paper, we limit related work to the field of preprocessing techniques in the process mining domain. We refer to [17] for an overview of different preprocessing techniques in data mining.

[7] indicates many quality issues for event logs. In [23], the authors identify some event log imperfection patterns that reduce the trustworthiness of process mining results. Moreover, [25] outlines typical data quality problems in event data and possible approaches to tackle them. however, to use most of these

approaches, we need to have business knowledge of the underling process of the event log.

Many process discovery algorithms, e.g., [20, 2], were designed to be able to handle infrequent behavior in event data and improve the quality of discovered process models. But, these filtering methods are tailored towards the internal working of the corresponding algorithms and they are not able to be used for general-purpose event log preprocessing. Besides, they typically focus on a specific type of behavior, e.g, incompleteness. There are some research has been done to improve the quality of process discovery algorithms by preprocessing the event log. [24] proposes to filter out chaotic activities to have process models with higher quality. In [9], the authors propose to use an anomaly free automaton to remove infrequent behavior. Furthermore, it is also recommended to filter out process instances which contain infrequent behavior from event data using probabilistic [11] and sequence mining [12] approaches. Moreover, in [13, 14], we propose modifying infrequent behavior to more general ones in each process instance instead of removing them. In all of these approaches, the improvement in quality of discovered models is measured by F1-Measure, i.e., the combination of *precision* and *fitness* [8]. In [10], a prototype selection approach based on a clustering algorithm is used to improve F1-Measure and simplicity of discovered process models. Moreover, [21] proposes to consider a sample of traces to make it feasible to clustering traces of large real event logs.

There is also some research has been done to improve the performance of process mining algorithms and providing an approximation of process mining results. In [5], the authors recommend a statistical trace-based sampling method to decrease the discovery time and memory footprint. Furthermore, [6] recommends a trace-based sampling method specifically for the Heuristic miner. Likewise, in [15], we analyze random and biased sampling methods with which we are able to adjust the size of the sampled data for process discovery. Moreover, some research has been done to approximate the alignment value by preprocessing event logs. [4] proposes to statistically sample the event log and applying the conformance checking algorithm on the sampled data. In [16], we propose that if just a part of behavior in an event log is used for conformance checking, we are able to have a suitable approximation of it faster. There is also some research has been done on using preprocessing methods that help us to have an approximation of performance analysis [5].

## 3 Research Approach

As shown in Figure 1, We consider preprocessing methods a function (i.e., $\rho$) that receives an event log (i.e., $L$) and returns a preprocessed event log (i.e., $L'$). In this way, a process mining method (i.e., $PM$) can be applied directly on the preprocessed event log without need to modify it. Here, we consider that $PM$ has only one input (i.e., $L$), but in reality it may have more inputs. To evaluate the efficiency of the preprocessing methods, we are able to consider the performance of mining methods or quality of results. To compute performance improvement,

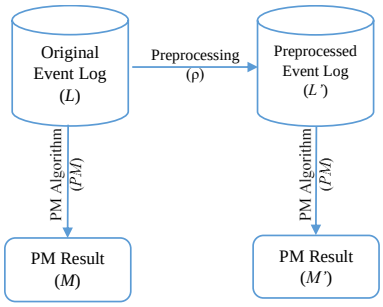


**Fig. 1.** The methodology of using preprocessing methods in process mining.

we consider the time of $PM$ on $L$ and compare it with its computation time on $L'$ plus preprocessing time. However, for some applications, we can ignore the preprocessing time. Note that, we compute the quality of both $M$ and $M'$ based on the original log. In general, with preprocessing methods, we reduce the complexity of event data by preprocessing methods. The complexity of an event log depends on many factors such as number of traces, number of unique trace-variants, number of activities and average length of traces in it. Therefore, the research goals of preprocessing functions that are presented in this research are 1- improving the quality of $M'$ compared to $M$ respect to $L$ 2- Reducing the required time of $PM$ on $L'$ when $M' \sim M$.

Figure 2 shows different preprocessing approaches that can be used to reduce the complexity and size of event logs. To simplify the concept, we consider an event log as tabular data that rows correspond to traces (or process instances) and columns show the activities. Note that for each approach, it is possible to have different preprocessing methods. The first approach that covers the majority of preprocessing methods aims to select some rows and put them as they are in the preprocessed event log. In other words, they focus on selecting some process instances of the original event log and putting them in the preprocessed event logs. Depends on the goal of preprocessing, some techniques try to not select process instances with outlier behavior (e.g., [11, 12, 18] and some others aim to select some representative process instances (e.g., [4, 15, 10, 16]).

Moreover, to improve the performance of process mining results (and sometimes their quality and at the same time) it is also possible to select some activities and project event logs on them (e.g., [24]). Note that many process mining algorithms are performing linear on the number of process instances in the given event log, but they perform exponential on the number of activities [19]. However, by removing activities, we sometimes add new behavior that does not exist in the original event log. For example, by removing $b$ from sub-sequence $\langle a, b, c \rangle$, we implicitly say that there is a direct relation between $a$ and $c$ that does not exist in the original sub-sequence.

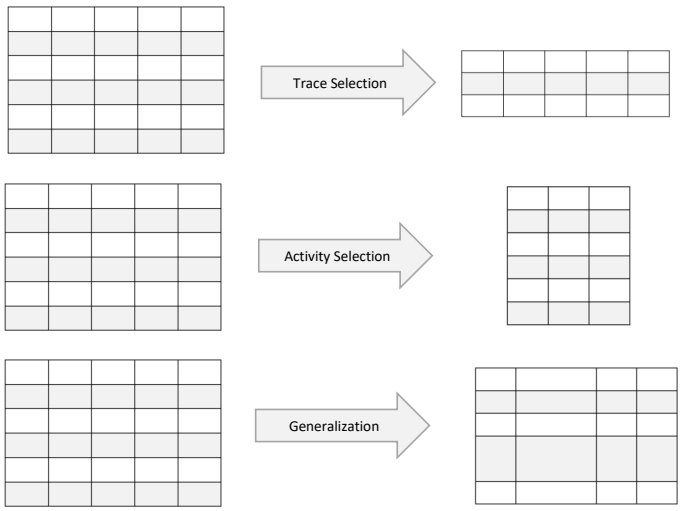


**Fig. 2.** Different event data preprocessing approaches.

Finally, it is also possible to merge unique process instances or activities to more general ones. In this way, we reduce the complexity of event logs by decreasing the uniqueness of behavior in it. We are able to apply this approach on just unique trace-variants level, just activity level, and on both of them.

Let $L \in \mathcal{B}(\mathcal{A}^*)$ be the original event log that is a multiset of sequence of activities where $\mathcal{A}$ is the set of activities in this event log. We define that $L' = \rho(L)$ is a preprocessing function where $L' \in \mathcal{B}(\mathcal{A}'^*)$ is the preprocessed event log. For trace selection approach, we have $L' \subseteq L$. Therefore, using this approach, we do not add any new behavior to the event log. However, in activity selection approach, we have $L' \in \mathcal{B}(\mathcal{A}'^*)$ where $\mathcal{A}' \subseteq \mathcal{A}$. Moreover, for generalization approach, it may exist some $a' \in \mathcal{A}'$ where $a' \notin \mathcal{A}$ and some traces $\sigma' \in L'$ where $\sigma' \notin L$. It should be noted that a preprocessing method can provide a hybrid approach that uses two or more of the mentioned approaches.

## 4 Current Results

In this section, we bring some results of current preprocessing functions considering the defined research goals. For details of the experiments, please see the corresponding reference. Note that all the event logs that are used for these experiments are real event logs which belong to different fields, from health-care to insurance[1]. Table 1 shows how by providing different preprocessing functions we are able to improve results of process discovery algorithms. In this table,

[1] The event logs are accessible from https://data.4tu.nl/repository/collection:event_logs.

| | Nothing | | | | | Sampling | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Log | Fitness | Precision | F1-measure | Model Size | Cardoso | Fitness | Precision | F1-measure | Model Size | Cardoso |
| BPIC-2012 | 1.00 | 0.12 | 0.21 | 27.75×35.75×515 | 224 | 0.88 | 0.24 | 0.37 | 24.7×170.4×27.7 | 133 |
| BPIC-2018-Dept. | 1.00 | 0.83 | 0.90 | 9×9×37.5 | 15 | 1.00 | 0.96 | 0.98 | 7.9×25.1×7.4 | 10 |
| BPIC-2018-Insp. | 1.00 | 0.13 | 0.23 | 19×26.25×311 | 150 | 0.96 | 0.37 | 0.51 | 17.0×54×21.8 | 90 |
| BPIC-2018-Ref. | 1.00 | 0.91 | 0.95 | 9.5×11×36 | 18 | 1.00 | 0.93 | **0.96** | 8.4×18.7×8.5 | 12 |
| BPIC-2019 | 1.00 | 0.36 | 0.53 | 43.5×46×1242.25 | 367 | 0.98 | 0.60 | 0.73 | 32.8×86.5×37.4 | 325 |
| Hospital | 1.00 | 0.39 | 0.57 | 20.75×23.5×410.5 | 111 | 0.98 | 0.59 | 0.72 | 17.3×81.9×16.8 | 66 |
| Road | 1.00 | 0.53 | 0.69 | 15×17.25×134.5 | 67 | 0.91 | 0.80 | 0.84 | 13.5×57.8×13.7 | 36 |
| Sepsis | 1.00 | 0.20 | 0.34 | 20×30.25×389.5 | 195 | 0.94 | 0.39 | 0.53 | 16.7×118.7×19.5 | 76 |

| | Statistical | | | | | Prototype Selection | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Log | Fitness | Precision | F1-measure | Model Size | Cardoso | Fitness | Precision | F1-measure | Model Size | Cardoso |
| BPIC-2012 | 1.00 | 0.12 | 0.22 | 27.7×36.1×572.9 | 232 | 0.75 | 0.74 | **0.65** | 24×26×170.4 | 81 |
| BPIC-2018-Dept. | 1.00 | 0.98 | **0.99** | 8.9×8×32.3 | 12 | 1.00 | 0.91 | 0.95 | 7.8×7.9×25.1 | 12 |
| BPIC-2018-Insp. | 1.00 | 0.16 | 0.28 | 18.9×24.6×286.7 | 125 | 0.88 | 0.64 | **0.68** | 13×14×54 | 0.1 |
| BPIC-2018-Ref. | 1.00 | 0.88 | 0.94 | 9.1×9.3×32.3 | 15 | 0.96 | 0.95 | 0.95 | 7.8×7.8×18.7 | 9 |
| BPIC-2019 | 1.00 | 0.52 | 0.68 | 42.3×52.9×1350.9 | 471 | 0.89 | 0.84 | **0.82** | 13×15.5×86.5 | 19 |
| Hospital | 1.00 | 0.44 | 0.61 | 20.7×22×381.1 | 109 | 0.87 | 0.86 | **0.84** | 14×12.45×81.9 | 30 |
| Road | 1.00 | 0.61 | 0.76 | 15×15.7×100 | 47 | 0.86 | 0.89 | **0.85** | 12.95×12.2×57.8 | 28 |
| Sepsis | 1.00 | 0.22 | 0.35 | 20×30.6×456 | 206 | 0.81 | 0.68 | **0.65** | 15.8×18.6×118.7 | 52 |

**Table 1.** Average values of process model quality criteria measures per preprocessing method for different event logs and process models [10]. Cardoso and Mode Size measure the complexity of process model that the later one indicates the number of transitions, places, and arcs in a Petri net.

*Nothing* refers to not using any preprocessing algorithm (i.e., our baseline). On the other hand, *Sampling* [15], *Statistical* [5], and *Prototype Selection* [10] are different preprocessing functions that all of them work based on the trace selection approach. To compute fitness and precision, the original event logs are used. It is shown that using preprocessing methods leads to improve the quality of discovered process models. This goal is usually achieved by scarifying a little in fitness and increasing a lot in precision.

In Figure 3, we show that how by applying different sampling process methods, we are able to improve the performance of process discovery algorithms. The red dotted line shows the case that no preprocessing method is used. The y-axis indicates how many times the process discovery algorithms are faster using different preprocessing methods. Here, we consider some different instance selection methods (i.e., [15, 5]). Results indicate that by applying these preprocessing methods we can improve the performance of process discovery algorithms. In Table 1 and in the cited papers in more detail, it is shown those process models that are discovered using some of these preprocessing methods have a higher quality compared to the case that the original event logs are used.

Using preprocessing is not limited to just process discovery algorithms. In Figure 4, it is shown how by using different preprocessing methods we can improve the performance of conformance checking. The red dotted line indicates the required time for the normal conformance checking method. Moreover, the y-axis shows how much faster the alignment value is computed using the preprocessing methods. It is indicated in [16] that most of the proposed methods are able to provide accurate conformance approximation values for these event logs.

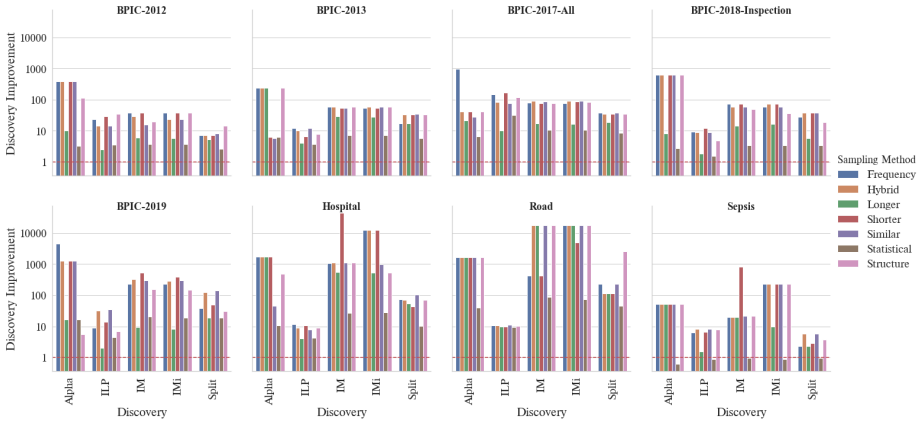


**Fig. 3.** Improvement in performance of process discovery using instance selection methods that are proposed in [15].

## 5 Challenges

There are the following main challenges in front of this research.

- How to measure the quality of process mining results in a quantifiable way? For instance, much of the research uses F-Measure (i.e., the combination of fitness and precision) to compare different process models. We think that using this measure is not accurate enough, specifically because it contains the precision metric. For example, the split miner [2] usually returns a process model with high F-Measure, but with also a high complexity [3]. Moreover, for some other process mining results, the comparison is even more challenging because there is no specific measure for them. As an example, it is challenging to compare performance analysis results that are gained by different prepossessing methods. It is also challenging to measure the accuracy of some approximated process mining results.
- Finding the best preprocessing parameters setting. Most of preprocessing methods have some parameters that by changing them we will have different results. The best setting of preprocessing parameters is various when we deal with different event logs. Moreover, sometimes there is a trade-off between performance improvement and the quality of process mining results. Consequently, adjusting these parameters is difficult and sometimes needs to be done using a trial and error method. Therefore, it is worth to at least reduce the range of parameters' values based on the characteristics of the given event log.

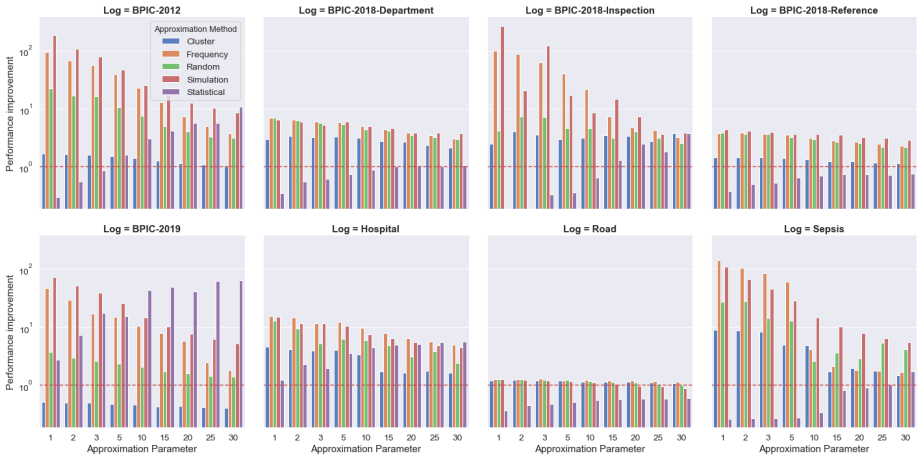


**Fig. 4.** Improvement in performance of conformance checking computation with consideration of preprocessing time for different preprocessing methods [16].

## 6 Conclusion

In this work, we explain that many process mining algorithms have difficulties dealing with real large/noisy event data. To tackle these problems, we aim to provide some preprocessing functions that reduce the complexity of event logs. At the same time, the results of process mining algorithms on the preprocessed event logs should be close to the original ones. We discuss some related work in this area and provide three general approaches to preprocess event logs. Our preliminary results show that the trace selection approach is able to improve the performance/quality of process mining results. We plan to develop new event log preprocessing methods that work based on activity selection and generalization approaches.

## Acknowledgement

This research is supervised by Prof. Wil van der Aalst and Dr. Sebastiaan van Zelst. We thank the Alexander von Humboldt (Avh) scholarship for funding this work.

## References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer Berlin Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Polyvyanyy, A.: Split miner: Automated Discovery of Accurate and Simple Business Process Models from Event Logs. Knowledge and Information Systems pp. 1–34 (2019)

3. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Maggi, F.M., Marrella, A., Mecella, M., Soo, A.: Automated discovery of process models from event logs: Review and benchmark. CoRR **abs/1705.02288** (2017)
4. Bauer, M., van der Aa, H., Weidlich, M.: Estimating process conformance by trace sampling and result approximation pp. 179–197 (2019)
5. Bauer, M., Senderovich, A., Gal, A., Grunske, L., Weidlich, M.: How much event data is enough? a statistical framework for process discovery. In: International Conference on Advanced Information Systems Engineering. pp. 239–256. Springer (2018)
6. Berti, A.: Statistical sampling in process mining discovery. In: The 9th International Conference on Information, Process, and Knowledge Management. pp. 41–43 (2017)
7. Bose, R.J.C., Mans, R.S., van der Aalst, W.M.P.: Wanna improve process mining results? In: 2013 IEEE symposium on computational intelligence and data mining (CIDM). pp. 127–134. IEEE (2013)
8. Buijs, J.C., van Dongen, B., van der Aalst, W.M.P.: On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery. In: OTM, " On the Move to Meaningful Internet Systems". pp. 305–322. Springer (2012)
9. Conforti, R., La Rosa, M., ter Hofstede, A.: Filtering Out Infrequent Behavior from Business Process Event Logs. IEEE Trans. Knowl. Data Eng. **29**(2), 300–314 (2017). https://doi.org/10.1109/TKDE.2016.2614680
10. Fani Sani, M., Boltenhagen, M., van der Aalst, W.: Prototype selection based on clustering and conformance metrics for model discovery. arXiv preprint arXiv:1912.00736 (4), 471–507 (2019)
11. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Improving Process Discovery Results by Filtering Outliers Using Conditional Behavioural Probabilities. In: Business Process Management BPM Workshops, Barcelona, Spain. pp. 216–229 (2017)
12. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Applying sequence mining for outlier detection in process mining. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 98–116. Springer (2018)
13. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Repairing outlier behaviour in event logs using contextual behaviour. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. **14**, 5:1–5:24 (2018)
14. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Repairing outlier behaviour in event logs using contextual behaviour. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. **14**, 5:1–5:24 (2018). https://doi.org/10.18417/emisa.14.5, https://doi.org/10.18417/emisa.14.5
15. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: The impact of event log subset selection on the performance of process discovery algorithms. In: New Trends in Databases and Information Systems, ADBIS 2019 Short Papers, Workshops BBIGAP, QAUCA, SemBDM, SIMPDA, M2P, MADEISD, and Doctoral Consortium, Bled, Slovenia, September 8-11, 2019, Proceedings. pp. 391–404 (2019)
16. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.: Conformance checking approximation using subset selection and edit distance. arXiv preprint arXiv:1912.05022 (2019)
17. García, S., Luengo, J., Herrera, F.: Data preprocessing in data mining. Springer (2015)
18. Ghionna, L., Greco, G., Guzzo, A., Pontieri, L.: Outlier Detection Techniques for Process Mining Applications. In: ISMIS 2008. pp. 150–159 (2008)

19. Hompes, B., Verbeek, H., van der Aalst, W.M.P.: Finding suitable activity clusters for decomposed process discovery. In: International Symposium on Data-Driven Process Discovery and Analysis. pp. 32–57. Springer (2014)
20. Leemans, S.J., Fahland, D., van der Aalst, W.M.P.: Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In: BPI, pp. 66–78 (2014)
21. Lu, X., Tabatabaei, S.A., Hoogendoorn, M., Reijers, H.A.: Trace clustering on very large event data in healthcare using frequent sequence patterns. In: Business Process Management - 17th International Conference, BPM 2019, Vienna, Austria, September 1-6, 2019, Proceedings. pp. 198–215 (2019)
22. Pyle, D.: Data Preparation for Data Mining. morgan kaufmann (1999)
23. Suriadi, S., Andrews, R., ter Hofstede, A., Wynn, M.T.: Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. Information Systems **64**, 132–150 (2017)
24. Tax, N., Sidorova, N., van der Aalst, W.M.P.: Discovering more Precise Process Models from Event Logs by Filtering out Chaotic Activities. Journal of Intelligent Information Systems pp. 1–33 (2018)
25. Wynn, M.T., Sadiq, S.: Responsible process mining-a data quality perspective. In: International Conference on Business Process Management. pp. 10–15. Springer (2019)