Demystifying Noise and Outliers in Event Logs: Review and Future Directions

Agnes Koschmider[1], Kay Kaczmarek, Mathias Krause, and
Sebastiaan J. van Zelst[2,3]

[1] Group Process Analytics, Kiel University, Germany
ak@informatik.uni-kiel.de
stu96465|stu113088@mail.uni-kiel.de
[2] Fraunhofer Institute for Applied Information Technology,
Fraunhofer Gesellschaft, Germany
sebastiaan.van.zelst@fit.fraunhofer.de
[3] Chair of Process and Data Science, RWTH Aachen University, Germany

Abstract. Various process mining techniques exist, e.g., techniques that automatically discover a descriptive model of the execution of a process, based on event data. Whereas the premise of process mining is clear, i.e., as witnessed by the tremendous growth of the field, data quality issues often hamper the direct applicability of process mining techniques. Several authors have studied data quality issues in process mining, yet, these works primarily propose data pre-processing techniques. An overarching study of the nature of data quality issues, the types of available techniques, and the general possibilities of (semi)-automated outlier/noise detection methods is missing. Therefore, in this paper, we propose a first attempt to structure and study the field of outlier/noise detection in process mining and understand to what degree knowledge on noise and outliers from other domains could advance the process mining field. We do so by answering three central research questions, covering various aspects related to (semi)-automated outlier/noise detection.

1 Introduction

Process mining [1] techniques are able to, largely automatically, derive valuable insights in the execution of a process, based on recorded *event data*. For example, techniques exist that automatically discover a process model based on event data, i.e., referred to as *automated process discovery algorithms* [2]. The premise of process mining is promising, yet, several practical issues, primarily related to *outlier and noise detection*, hamper the direct application of process mining techniques on the recorded event data. As such, various techniques to pre-process and filter event data have been proposed in process mining, e.g., [3,4,5]. However, a clear overview and categorization of these techniques, as well as the concepts of *noise* and *outliers* have not been proposed.

In statistics, outliers are defined as "*high measurements where the value is some standard deviation above the average*" [6]. In data engineering, outliers, commonly referred to as "*anomalies*", refer to "*something that is out of range*". This can, on the one hand, point to insignificant data or, on the other hand, to interesting and useful information about the underlying system [7]. Hence, distinguishing the essence of outliers in terms of undesired or unwanted behavior versus surprisingly correct and informative data is of particular interest for the quality of process discovery. However, studies on the interrelation between outliers and noise have received little interest. We consider outliers in event logs as *divergent data* that is out of range of behavior that we expect, while noise is "semantic", i.e., it refers to, e.g., erroneous data recordings. Both outlier and noise detection are essential for process discovery since they might negatively influence the usefulness of the discovered process model [8]. For example, incorrect logging of event positions causes problems for process discovery algorithms to discover the correct control flow (i.e., relationships between events may be inferred that do not exist in reality). Consequently, a clear understanding of noise and outliers bridges the gap between detecting errors in event data recordings versus unexpected or even unwanted, yet, correclty logged behavior.

While several approaches exist to handle outliers in event logs and to filter noise within the process discovery algorithms [4,9,10], we aim to understand to which degree the potential of outlier and noise detection techniques from other fields have been exploited for process mining. To provide an answer, we formulate the following three research questions: **RQ1**: What types of outlier and noise exist for event logs? **RQ2**: How are outliers and noise detected in event logs? **RQ3**: What are potential research directions for future research in outlier and noise detection techniques in process mining? Equipped with such knowledge, we aim to define an outlier and noise detection model that quantifies outliers and noise in event logs.

Against this background, this paper is structured as follows. Section 2 classifies outlier, noise, and "normal" behavior in an event log aiming to answer **RQ 1**. Relying on this classification, a literature search has been conducted with the purpose of addressing **RQ 2**. The results are summarized in Section 3. To further understand the capacities of noise and outlier techniques for event logs, corresponding tools were analyzed. Based on the literature search results and the tool analysis, potential research directions are formulated and summarized in Section 5 referring to **RQ 3**. The paper ends with a conclusion.

2 Noise vs. Outliers

Techniques for outlier detection have been suggested to a large extent for data analysis in general [7], whereas a clear conceptualization of outlier and noise detection for event logs is lacking. Instead, noise and outliers are often used synonymously. Fig. 1 illustrates how outliers and noise in an event log can be considered and differentiated from normal behavior. An outlier in event data, is a behavior that is out of range of the behavior that we expect. It is something that differs considerably from all or most other behavior. According to Fig. 1 these are points laying outside the thin, dotted rectangle. Noise is any undesirable or

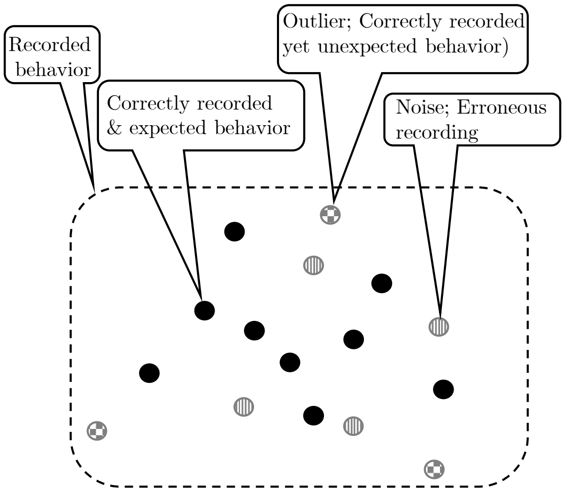
Recorded behavior
Correctly recorded & expected behavior
Outlier; Correctly recorded yet unexpected behavior)
Noise; Erroneous recording

Fig. 1: Difference between outliers, noise and normal behavior in an event log.

unwanted value of a case or event attribute. Behavior without any outlier or noise is considered "normative behavior" and is represented as a black dot in Fig. 1.

To understand outliers in event logs, we use an existing classification from the literature [11] and map it for our purpose on outliers in event logs:

- *Point as outlier*: this is the simplest type of ouliers and plenty of literature exist how to detect them, for an overview we refer to [12]. In the context of an event log *point as outlier* would be a trace with activities significantly deviating from the rest. E.g., in a log $L = \{\langle A, B, C, E, F\rangle^{50}, \langle A, B, C, D, E, F\rangle^{100}, \langle A, A, A, B, C, E, F\rangle^{2}\}$ the third trace is the only trace with three As and points to an outlier. Note that although this trace is classified having an outlier, this outlier might point to interesting observations.
- *Context as outlier*: this category of outliers, also referred to as conditional anomalies, requires a notion of context. A contextual outlier is given if an observation is uncommon in a certain context but not unexpected in another context. A contextual outlier is a trace that significantly depends on a context dimension. To structure context dimensions for our purpose, we apply context dimensions as they have being defined for conceptual models [13], which are: personal & social, task, environmental and spatial-temporal. For instance, the process for oncology in the daytime consists of 14 activities while the process for intensive care in the night has 8 activities with two loops. In this example time is a contextual factor (it refers to the spatial-temporal dimension) and thus traces with 8 activities are annotated with spatial-temporal attribute.
- *Subsequence as outlier*: in the context of an event log a subsequence outlier means that a subset of the trace deviates significantly from the rest, even if

the individual activities in the subset may not be outliers. For example in a log $L = \{\langle A, B, C, E, F, G\rangle^{30}, \langle A, B, C, D, E, F, G\rangle^{25}, \langle A, B, C\rangle^{2}\}$ the last trace is an outlier since there are only a few traces of it although the behavior A, B, C is a regular behavior.

Common techniques to detect outliers are density-based or distance-based clustering [14]. These algorithms build a model (e.g. a statistical, probabilistic model) describing the normal behavior, and consider as outliers all data points which deviate from this model [14]. In the context of event logs also sequence-based anomaly detection, contiguous subsequence-based anomaly detection and pattern frequency-based anomaly detection are common techniques, since they consider particular aspects of event logs for outlier detection [15].

To understand and classify noise, we browsed the literature on data quality for databases and machine learning [16,17,18,19] and event log imperfection patterns [20]. Accordingly, we adopt the classification of attribute and class noise and map it for our purpose:

- *Attribute noise*: Arises when an imprecision, incompletion or an error is introduced to one or more attributes. Attribute noise can be totally unpredictable i.e., random, or simply a low variation with respect to the correct value [16]. In the literature the following types of attribute noise are distinguished [16]: (1) erroneous attribute values, (2) missing or don't know attribute values and (3) incomplete attributes or don't care values [16]. An event log might contain erroneous attribute values due to a logging error that recorded identical timestamp for different events (see Table 1). Missing attribute values might arise due to e.g., faults in sensor devices and they are shown by a missing entry in an event log. Incomplete attributes might occur due to irregularities in sampling like that the data may not be available or it was not considered important.
- *Class noise*: this category refers to the semantics of attributes (i.e., mislabeled or contradictory activity labels). The event log imperfection patterns [21] describe examples for class noise like the patterns Synonymous Labels and Homonymous Label.

Figure 2 shows in which steps attribute and class noise can be found in the process from raw (sensor) event data to the discovered process model and also where outliers affect this process. This figure might lay the foundation to conceptualize a holistic detection technique for outliers and noise. Although attribute and class noise can already be found in the raw (sensor) event data, the classification if this noise is completely random or not is determined at event log level. Class noise is also detected when abstracting or aggregating events to activities since the semantics of attributes is disambiguated at this stage. Outliers are detected either at the aggregation level or during process discovery.

To understand how outliers and noise have been addressed for process mining, a literature search was conduced. The results are summarizes in the next section.

Table 1: An example event log, adapted from [22], with attribute noise (highlighted in red) due to identical timestamps and missing values that were not recorded for resources.

Case id	*Timestamp*	*Activity*	*Resource*	*Transactional*	*Cost*	···
12373	30-7-2019 11.12	register request	Bas	complete	50	···
12374	30-7-2019 11.32	register request	–	start	50	···
12374	30-7-2019 11.44	register request	Agnes	complete	50	···
12373	30-7-2019 11.44	check ticket	–	start	100	···
⋮	⋮	⋮	⋮	⋮	⋮	...

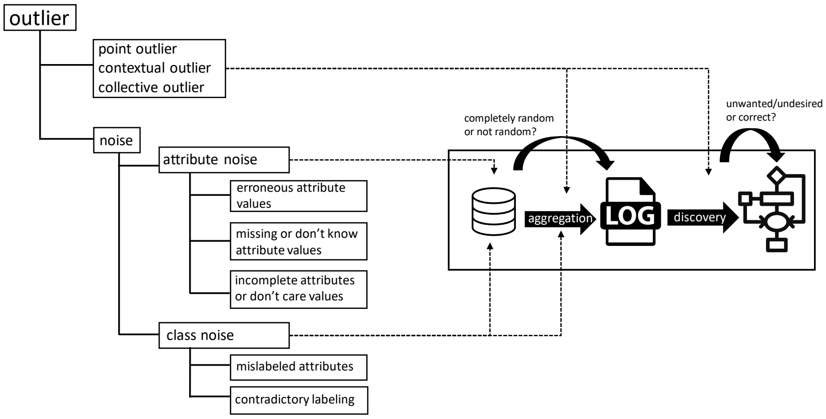
outlier
point outlier
contextual outlier
collective outlier
noise
attribute noise
erroneous attribute values
missing or don't know attribute values
incomplete attributes or don't care values
class noise
mislabeled attributes
contradictory labeling
completely random or not random?
unwanted/undesired or correct?
aggregation
LOG
discovery

Fig. 2: Classification of outliers and noise and its relevance for the steps from raw data to the discovered process model.

3 Literature Search

This section describes how the literature search was conducted. We searched the ACM, IEEE and Springer research databases. We also used Google Scholar to find appropriate literature by browsing the citations of related publications already found in the scientific databases. Second, we conducted a backward search to find more appropriate publications cited in papers of the first search round. To find synonyms for the term *outlier* we checked all related papers found by Springer having this term in the title. The result list was cross-checked with related papers on event log quality [20,23] and one highly cited paper on outlier detection for process discovery [3]. We used the following query to identify appropriate publications:

```
{(‘‘process mining’’ OR ‘‘event log’’)  AND
(‘‘outlier’’ OR ‘‘anomaly’’ OR ‘‘infrequent behavior’’ OR ‘‘noise’’
OR ‘‘abnormal behavior’’) AND ‘‘process mining’’}
```

Table 2: Type of Outlier and Noise techniques suggested for event logs.

Outlier		
	Point	chaotic activity [26] spurious event [10,4,27]
	Contextual	personal & social (e.g, resource) [28] task (e.g., control-flow): [5,29,9,30,31,32,33,25] environmental (e.g., training data): [34,35,36] spatial-temporal (e.g., time): [31]
Noise		
	Attribute Noise	erroneous attribute values: [37,25,38,24] missing attributes values [39] incomplete attribute values [40,3,25]
	Class Noise	synonymous label [38,41,42,24] homonymous label [24].

Eventually, we ended up with 24 relevant publications. The 24 publications were published in the following years: 2008 (4.5%), 2015 (9.1%), 2017 (13.6%), 2018 (31.8%), 2019 (31.8%), 2020 (9.1%). These numbers indicate a recent increasing interest in the topic.

3.1 Literature Result: Type of Outliers and Noise

Table 2 shows the results of the literature search. Referring to **RQ1** (What types of outlier and noise exist for event logs?) the following observations can be given. The main focus has been put on control-flow aspects like probability and frequency of traces. Some works exist on point as outlier detection. Only one paper has been found that detects attribute and class noise, see [24] and one paper [25] addressing attribute noise and context as outlier. No paper was found for subsequence as outlier.

3.2 Literature Result: Outlier Detection Techniques

To answer **RQ2** (How are outliers and noise detected in event logs?) we analyzed the related literature with respect to the detection techniques used to identify noise and outliers. Table 3 summarizes the results. It shows that unsupervised learning is the most common technique. According to our analysis only one approach (see reference [25]) discusses the characteristics of outliers (i.e., is this undesired, unwanted behavior vs. surprisingly correct and informative data).

4 Analysis of Tools for Noise and Outlier Filtering for Event Logs

The intention of this analysis is to identify the efficiency of available tools and to understand if noise and outliers are kept properly apart. We first narrowed our analysis on noise detection in event logs. For this purpose, we generated an event

Table 3: Outlier detection techniques used for process mining

Article	**Type**	**Year**	**Type of Supervision**			**Technique**	
			Superv.	Semi-s.	Unsuperv.	Method	Explain?
Ghionna et al. [34]	-O-	2008			x	Distance	
Wang et al. [41]	-N-	2015			x	Branch and Bound	
Cheng et al. [36]	-O-	2015	x			Rule based	
Nolle et al. [35]	-O-	2016			x	Deep Learning	
Conforti et al. [3]	-N-	2017			x	Log Automaton	
Chapela-Campa et al. [32]	-O-	2017			x	Pattern	
Mannhardt et al. [33]	-O-	2017			x	Density	
Fani Sani et al. [5]	-O-	2018			x	Rule based	
Fani Sani et al. (b) [29]	-O-	2018			x	Covering Probability	
Fani Sani et al. (c) [9]	-O-	2018			x	Density	
van Zelst et al. [10]	-O-	2018			x	Probabilistic Automata	
Conforti et al. [37]	-N-	2018			x	Automata	
Nolle et al. [31]	-O-	2018		x		Deep Learning	
Tax et al. [26]	-O-	2019			x	Entropy,Statistical	
Böhmer et al. [28]	-O-	2019			x	Rule based	
Sun et al. [27]	-O-	2019			x	Density	
Sadeghianasl et al. [38]	-N-	2019		x		Density	
Chapela-Campa et al. [30]	-O-	2019			x	Pattern, Abstraction	
Nguyen et al. [39]	-N-	2019			x	Machine Learning	
Niels et al. [25]	-O-N-	2019			x	Heuristics	x
Sadeghianasl et al. [42]	-N-	2019			x	MST clustering	
Sarno et al. [40]	-N-	2020			x	Rule based	
van Zelst et al. [4]	-O-	2020			x	Probabilistic Automata	

log with 1000 traces based on the Petri net shown in Fig. 3[4]. To add noise to the event log, we use the ProM[5] plugin ”Add Noise to log filter” with a noise threshold of 20%. A manual inspection, however, shows that the changed event log has only 135 out of 1000 noise-free traces (=13,5%). Due to this inappropriate result,

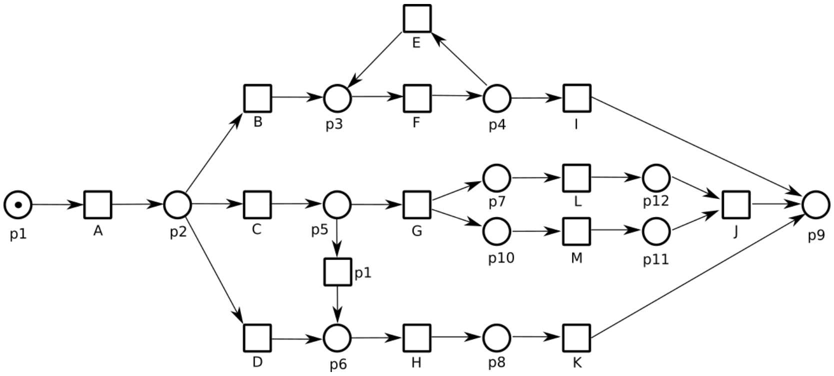

Fig. 3: Petri net used to generate a synthetic event log.

[4] To generate the event log we used the Petri Net-based Event Log Generator `http://processmining.be/loggenerator/`

[5] `https://www.promtools.org/doku.php`

we decided to manually add (attribute) noise[6]. Based on the Petri net shown in Fig. 3 we generated a log with $L = \{\langle A, D, H, K\rangle^{68}, \langle A, C, N, H, K\rangle^{22}, \langle A, C, G, L, M, J\rangle^{12}, \langle A, C, G, M, L, J\rangle^{3}, \langle A, B, F, I\rangle^{25}, \langle A, B, F, E, F, I\rangle^{4}, \langle A, B, F, E, F, E, F, I\rangle^{1}\}$. To understand whether available tools also detect outliers and not only noise, we inserted varying trace frequency and control-flow errors into the event log. Additionally, we assign resources to activities, which is relevant when analyzing noise like missing attribute values or incomplete assignment of resources. For this purpose, we assign ResGr1 to activity A, ResGr2 to activities B,E,F,I, ResGr3 to activities C, G, L, M, J and ResGr4 to activities D, N, H, K. Normal behavior are all traces that can be replayed on the Petri net of Fig. 3. Besides, we define two control-flow rules: 1) activity E may only be performed a maximum of three times, 2) the order in which activities M and L are performed is not important. However, activity G must precede both activities and J must succeed. Abnormal behavior is any behavior that deviates from our defined behavior according to our Petri net. For this event log we manually inspected these four tools:

- The process mining tool Disco[7]: here filtering of specific attributes is not possible nor filtering by frequency within a trace or the frequency of subsequently followed activities.
- Filter events based on attribute values [5]: this tool only allows to filter events from traces, but not the incorrectly assigned resources.
- Filter Event Log[5]: although this tool has plenty of noise filtering techniques, some errors were found in this tool. For instance, filtering the order in which activities have to be executed does not work appropriately.
- Filter/Edit Attributes of an Event Log[5]: the filtering technique is limited to renaming of attributes rather than noise or outlier detection.

Summarizing, our analysis shows that available noise filtering tools do not appropriately filter noise. Although they aim to identify outliers (i.e., point as ouliers) there is ample potential to improve the filtering techniques e.g., in terms of erroneous order of activities. We are convinced that our classification of outliers and noise could be seen an appropriate foundation for revision. In future we plan to compare available tools for outlier detection with emphasis on control-flow in order to give more insights into the quality of the detection techniques.

5 Future Research Directions

Based on our different analysis, we see the following future research directions responding to **RQ3**:

Correlation between noise and outliers on process quality: No paper exists analyzing the influence of noise and outliers on process discovery (i.e., Is it more beneficial, first to find noise and then outliers or vice versa? How does

[6] Available tools do not resolve synonyms nor homonyms. Therefore we restricted our analysis only to attribute noise.

[7] `https://fluxicon.com/disco/`

different attributes impact the process quality? or How does attribute and class noise detection impact each other?) In terms of data quality and classification accuracy for machine learning, plenty of experiments were conducted to analyze the mutual impact of attribute and class noise [16,18,19]. Related literature shows that attribute noise depends on the correlation between attribute and class noise. The higher the correlation, the more negative impact the attribute noise may bring [16]. Noisy attributes that have a weak correlation with class label could be eliminated, while noisy attributes with a high correlation with the class label may be subject to data cleansing. Our analysis shows that such analysis have not been conducted for event logs. To provide a solution, attributes could be ranked according to their sensitivity from most to least noisy [18] and then to apply polishing techniques instead of noise filtering for highly noise sensitive attributes [19].

Explainability: Our analysis shows that only little attention has been given to the explainability of outlier and noise detection. As shown in Table 3 existing approaches do not discuss the characteristics of outliers like interesting vs. unwanted behavior, nor if the attribute noise is random. However, this classification is essential for appropriate filtering. A thorough analysis of explainability, also calls for experimental settings on event log quality of the complete data pipeline. This calls for a holistic approach evaluating the impact of noise and outlier and its mutual impact on process discovery.

6 Related Work

A partial view on outlier detection for process mining has been addressed by [43]. This paper, however, mainly focuses on outlier detection in business process runtime behavior and sets a different focus on this topic. Complementary to our paper are approaches on data quality issues in event logs [20,23]. The data quality frameworks defined in [20,23] classify quality issues, however, they do not provide guidance as how to discover noise and outliers for process mining, nor do they provide a mechanism to discuss to which extent an event log is affected by noise and outliers (i.e., how is the quality of an event log affected by attribute noise?). Quality issues in an event log can be identified by event log imperfection patterns `http://www.workflowpatterns.com/patterns/logimperfection/`, which are mainly related to noise and disregard outliers.

Some process discovery algorithms like the Inductive Miner has embedded filtering mechanisms to deal with some types of outliers. According to [44] infrequent behavior is considered as paths that are taken infrequently, or traces that only differ by occurrence of infrequent activities. The directly-follows graph (DFG) is filtered until it only contains most frequent edges or the mainstream behavior (often called as happy path). When applying this approach, however, it has been identified that the DFG may be misleading [45]. To overcome limitations of the directly-follows graph in terms of appropriately filtering infrequent behavior, the eventually-follows graph (EFG) has been suggested [44].

To the best of our knowledge this is the first literature review on outlier and noise detection for process mining.

7 Conclusion and Future Directions

It is generally acknowledged that outlier detection is a challenging task. Anyway, handling outliers is an essential task in order to identify desired vs. correct deviations. Within the field of process mining, data quality, i.e., the presence of outliers and/or noise, is one of the most urgent problems hampering the direct application of process mining techniques on event data. Whereas a significant amount of research has been conducted toward (algorithmic) techniques for outlier/noise identification and removal, an overall viewpoint on the matter is lacking. In this paper, on the basis of existing literature on general data quality issues, we provide a structured qualification of the notions of outliers and noise in event data. Furthermore, we provide an overview and classification of the different existing techniques developed for the purpose of outlier/noise detection and removal for process discovery. Future research avenues in this field are (1) discussions about the characteristics of outliers (i.e., undesired, unwanted behavior vs. surprisingly correct and informative data), (2) holistic approach to analyze attribute and class noise, while class noise has received little attention yet, (3) sophisticated detection techniques for event logs (e.g., filtering by specific attributes, their frequency within a trace or the frequency of subsequently followed activities).

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. 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. IEEE Trans. Knowl. Data Eng. **31**(4) (2019) 686–705
3. Conforti, R., Rosa, M.L., ter Hofstede, A.H.M.: Filtering out infrequent behavior from business process event logs. IEEE TKDE **29**(2) (2017) 300–314
4. van Zelst, S.J., Sani, M.F., Ostovar, A., Conforti, R., Rosa, M.L.: Detection and removal of infrequent behavior from event streams of business processes. Inf. Syst. **90** (2020) 101451
5. Fani Sani, M., van Zelst, S., Aalst, W.: Applying sequence mining for outlier detection in process mining: Confederated international conferences: Coopis, c&tc, and odbase 2018, proceedings, part ii. (10 2018) 98–116
6. Freedman, D.: Statistical Models : Theory and Practice. Cambridge University Press (August 2005)
7. Ord, K.: Outliers in statistical data, 3rd edition, (john wiley & sons, chichester). International Journal of Forecasting **12**(1) (1996) 175–176
8. Ghionna, L., Greco, G., Guzzo, A., Pontieri, L.: Outlier detection techniques for process mining applications. Volume 4994. (05 2008) 150–159

9. Sani, M.F., van Zelst, S.J., van der Aalst, W.M.P.: Improving process discovery results by filtering outliers using conditional behavioural probabilities. In: BPM Workshops, Springer (2018) 216–229
10. van Zelst, S.J., Sani, M.F., Ostovar, A., Conforti, R., Rosa, M.L.: Filtering spurious events from event streams of business processes. In: CAiSE 2018, Proceedings. (2018) 35–52
11. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. **41**(3) (2009)
12. Gupta, M., Gao, J., Aggarwal, C.C., Han, J.: Outlier detection for temporal data: A survey. IEEE TKDE **26**(9) (2014) 2250–2267
13. Koschmider, A., Mannhardt, F., Heuser, T.: On the contextualization of event-activity mappings. In: BPM 2018 International Workshops. (2018) 445–457
14. Aggarwal, C.C.: Outlier Analysis. 2nd edn. Springer (2016)
15. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection for discrete sequences: A survey. IEEE Trans. on Knowl. and Data Eng. **24**(5) (2012) 823–839
16. Zhu, X., Wu, X.: Class noise vs. attribute noise: A quantitative study of their impacts. Artif. Intell. Rev. **22**(3) (November 2004) 177–210
17. SáEz, J.A., Galar, M., Luengo, J., Herrera, F.: Tackling the problem of classification with noisy data using multiple classifier systems: Analysis of the performance and robustness. Inf. Sci. **247** (October 2013) 1–20
18. Khoshgoftaar, T.M., Van Hulse, J.: Empirical case studies in attribute noise detection. IEEE Transactions on Systems, Man, and Cybernetics **39**(4) (2009) 379–388
19. Gupta, S., Gupta, A.: Dealing with noise problem in machine learning data-sets: A systematic review. Procedia Computer Science **161** (2019) 466 – 474 The Fifth Information Systems International Conference.
20. Dixit, P.M., Suriadi, S., Andrews, R., Wynn, M.T., ter Hofstede, A.H.M., Buijs, J.C.A.M., van der Aalst, W.M.P.: Detection and interactive repair of event ordering imperfection in process logs. In: CAiSE, Springer (2018) 274–290
21. Andrews, R., Suriadi, S., Ouyang, C., Poppe, E.: Towards event log querying for data quality. In: OTM 2018 Conferences, Springer (2018) 116–134
22. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining - literature review and taxonomy. Granular Computing (2020)
23. Bose, R.P.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). (2013) 127–134
24. Ziolkowski, T., Brandt, L., Koschmider, A.: Elogqp: An event log quality pointer. In: ZEUS 2021. Volume 2839 of CEUR Workshop Proceedings., CEUR-WS.org (2021) 42–45
25. Martin, N., Martinez-Millana, A., Valdivieso, B., Fernandez-Llatas, C.: Interactive data cleaning for process mining: A case study of an outpatient clinic's appointment system. (09 2019) 532–544
26. Tax, N., Sidorova, N., van der Aalst, W.M.P.: Discovering more precise process models from event logs by filtering out chaotic activities. J. Intell. Inf. Syst. **52**(1) (2019) 107–139
27. Sun, X., Hou, W., Yu, D., Wang, J., Pan, J.: Filtering out noise logs for process modelling based on event dependency. In: ICWS 2019, IEEE (2019) 388–392
28. Böhmer, K., Rinderle-Ma, S.: Mining association rules for anomaly detection in dynamic process runtime behavior and explaining the root cause to users. Information Systems (2019)
29. Fani Sani, M., van Zelst, S., van der Aalst, W.: Repairing outlier behaviour in event logs. In: BIS 2018. LNBIP, Springer (1 2018) 115–131

30. Chapela-Campa, D., Mucientes, M., Lama, M.: Simplification of complex process models by abstracting infrequent behaviour. (10 2019) 415–430
31. Nolle, T., Seeliger, A., Mühlhäuser, M.: Binet: Multivariate business process anomaly detection using deep learning. In: BPM 2018, Proceedings. (2018) 271–287
32. Chapela-Campa, D., Mucientes, M., Lama, M.: Discovering infrequent behavioral patterns in process models. In: BPM, Springer (2017) 324–340
33. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Data-driven process discovery - revealing conditional infrequent behavior from event logs. In: CAiSE 2017, Proceedings. (2017) 545–560
34. Ghionna, L., Greco, G., Guzzo, A., Pontieri, L.: Outlier detection techniques for process mining applications. In: Foundations of Intelligent Systems, Springer (2008) 150–159
35. Nolle, T., Seeliger, A., Mühlhäuser, M.: Unsupervised anomaly detection in noisy business process event logs using denoising autoencoders. In: Discovery Science, Springer (2016) 442–456
36. Cheng, H.J., Kumar, A.: Process mining on noisy logs — can log sanitization help to improve performance? Decision Support Systems **79** (2015) 138 – 149
37. Conforti, R., La Rosa, M., ter Hofstede, A.: Timestamp repair for business process event logs. Technical report, Technical report (2018)
38. Sadeghianasl, S., ter Hofstede, A.H.M., Wynn, M.T., Suriadi, S.: A contextual approach to detecting synonymous and polluted activity labels in process event logs. In: OTM 2019 Conferences, Springer (2019) 76–94
39. Nguyen, H.T.C., Comuzzi, M.: Event log reconstruction using autoencoders. In: ICSOC 2018 Workshops, Springer (2019) 335–350
40. Sarno, R., Sinaga, F., Sungkono, K.: Anomaly detection in business processes using process mining and fuzzy association rule learning. Journal of Big Data **7** (12 2020)
41. Wang, J., Song, S., Lin, X., Zhu, X., Pei, J.: Cleaning structured event logs: A graph repair approach. Proceedings - International Conference on Data Engineering **2015** (05 2015) 30–41
42. Sadeghianasl, S., ter Hofstede, A.H.M., Wynn, M.T., Suriadi, S.: A contextual approach to detecting synonymous and polluted activity labels in process event logs. In: OTM Conferences. Volume 11877 of Lecture Notes in Computer Science., Springer (2019) 76–94
43. Böhmer, K., Rinderle-Ma, S.: Anomaly detection in business process runtime behavior - challenges and limitations. CoRR **abs/1705.06659** (2017)
44. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from incomplete event logs. In: Application and Theory of Petri Nets and Concurrency, Springer (2014) 91–110
45. van der Aalst, W.: A practitioner's guide to process mining: Limitations of the directly-follows graph. Procedia Computer Science **164** (2019) 321 – 328 CENTERIS 2019.