Active Anomaly Detection for Key Item Selection in Process Auditing

Ruben Post[1], Iris Beerepoot[1], Xixi Lu[1], Stijn Kas[1], Sebastiaan Wiewel[1], Angelique Koopman[2] and Hajo Reijers[1]

[1]Information and Computer Sciences, Utrecht University, Utrecht, The Netherlands,
[2]Department of Accountancy, Tilburg University, Tilburg, The Netherlands

Abstract. Process mining allows auditors to retrieve crucial information about transactions by analysing the process data of a client. We propose an approach that supports the identification of unusual or unexpected transactions, also referred to as exceptions. These exceptions can be selected by auditors as "key items", meaning the auditors wants to look further into the underlying documentation of the transaction. The approach encodes the traces, assigns an anomaly score to each trace, and uses the domain knowledge of auditors to update the assigned anomaly scores through active anomaly detection. The approach is evaluated with three groups of auditors over three cycles. The results of the evaluation indicate that the approach has the potential to support the decision-making process of auditors. Although auditors still need to make a manual selection of key items, they are able to better substantiate this selection. As such, our research can be seen as a step forward with respect to the usage of anomaly detection and data analysis in process auditing.

Keywords: Process Mining, Domain Knowledge, Anomaly Detection, Auditing

1 Introduction

In the past years, it has become clear that data captured by information systems are relevant for auditors [1,2]. Process mining allows auditors to elicit behaviour from process data in the form of event logs derived from information systems of their clients [1, p. 32]. An event log is a collection of cases, where a case is a sequence of events performed in the context of a single process [1, p. 128]. As an event log collects behaviour captured by the information systems involved, it can be considered as an unbiased perspective on the client's processes [3]. Take, for example, an event log that contains loan offers made by a bank. The bank receives a customer request for a loan, asks for additional information until it has sufficient information, and finally decides to grant the loan or not. Without process data, the auditor does not know the steps taken before the loan was granted (i.e. the behaviour), while this behaviour could be instrumental in the auditor's decision to further investigate a particular loan offer.

Choosing which loan offer, or any other transaction of the client, to investigate further is also referred to as key item selection. Key items are specific transactions that auditors want to look further into by, for example, requesting additional documentation on the transactions because they might have a higher likelihood of containing a material misstatement (i.e. transactions that violate accounting and auditing standards [4, p. 374]). Currently, key items are selected based on the size of the transactions (i.e. the transactions with the highest monetary value), professional judgement, or by drawing a sample [5, Ch. 6]. Without detailed information about transactions, unusual or unexpected aspects such as the number of times a transaction has been declined and resubmitted, the total number of activities performed in the transaction, or the throughput time of the transaction, are largely ignored while selecting key items.

Anomaly detection algorithms can be used to detect exceptions, which can then be selected as key items. However, both supervised and unsupervised approaches may be unsuitable for practice because they either require a large amount of labeled training data or lack explainability [6–11]. More specifically, selecting the key items based on the underlying process data still requires domain knowledge [11]. Hence, active engagement of domain experts (i.e. auditors) is needed to detect exceptions. This leads to the following research question: *How can key item selection be supported using active anomaly detection on process data?* The research question is answered by structuring the identification of exceptions in process data in a three-step approach. In doing so, we contribute to the research field of process mining and auditing. By embedding domain knowledge in the identification of exceptions, the approach shows that the involvement of domain experts can be beneficial for both the domain experts' insight into the process of the client and the results of the approach itself. Additionally, because the approach provides a more detailed account of the transactions, the selection is better substantiated.

2 Related Work

2.1 Anomaly Detection

Anomaly detection approaches can be used to identify unusual or unexpected transactions in process data, also referred to as *exceptions*. Several anomaly detection approaches suitable for process data have been proposed [6–11]. Current approaches mostly use trained models to detect anomalies, like Ko et al. [10] and Pauwels et al. [9]. Using these approaches in practice is difficult because there is no labeled data available during audits, meaning these models cannot be trained. An alternative to training data could be a temporal holdout set where the data of the prior audit is used to train the model (i.e. the prior-year data is used to train the model to identify anomalies in the current-year data). However, if a temporal holdout set is used, concepts such as concept drift should be taken into account because the model might not know the difference between an anomaly and the introduction of a new process. An example of this is the recent COVID-19 pandemic, which coerced the digitisation of processes, introducing concept drift to the process data.

Other approaches such as those of Nolle et al. [8] and Böhmer et al. [7] use neural network-based autoencoders and Basic Likelihood Graphs to identify anomalies. This introduces complexity through the techniques they use, leading to both an increase in required processing power and unexplainability or incomprehensibility of the approach. This is problematic because it prohibits the domain expert to adequately substantiate their selection of key items.

In contrast to the other approaches, Schumann et al. [11] use low-complexity models that do not require training data to identify anomalies. Because they determine certain non-compliant patterns in the data beforehand and inject the data with these patterns, no training data is required. While the other approaches are evaluated through various performance metrics, only Schumann et al. [11] evaluate their technique with domain experts. The benefit of this type of evaluation is that is allows for domain experts to differentiate between real anomalies and cases that are considered compliant in practice. However, the rationale used by the domain experts is not explained, which brings the applicability and replicability of the approach in practice in question.

2.2 Active Anomaly Detection

Traditional anomaly detection approaches do not actively engage domain experts when identifying exceptions while the performance of anomaly detection approaches can potentially be improved by incorporating domain expert feedback. An example of a framework that allows using domain expert feedback is the Active Anomaly Detection (AAD) framework by Das et al. [12]. The AAD framework takes an ensemble model and assigns an initial weight to each individual model. The weight of a model influences how much it contributes to the anomaly score of a data point. A higher weight gives a model more influence on the anomaly score. After assigning an anomaly score to each case, a query budget B is defined and the instances with the top-B anomaly scores are labeled by domain experts. After each instance is labeled, the weights of the models are updated. The technical details of how the weights are updated are left out due to size limitations but can be found in [12].

To the best of our knowledge, anomaly detection approaches that actively engage domain experts are not currently used in practice. Furthermore, as mentioned above, selecting key items on the underlying process data still requires domain knowledge. AAD could provide domain experts the opportunity to embed their knowledge in the anomaly detection algorithm. However, because in this research process data is used, some additional steps need to be taken before the data is suitable the AAD framework. The reason the AAD framework is chosen is because it is written in a programming language compatible with current process mining techniques (i.e. Python).

2.3 Trace Visualisation

By using the AAD framework, domain knowledge is embedded in the assigning of the anomaly scores. It could however be that domain experts have different opinions on the label of a case. Hence, the information presented to domain experts should make clear **why** the presented case has a high anomaly score in an understandable and interpretable way.

Within process mining literature, different types of visualisations have been proposed, each of them serving different purposes [13]. According to Klinkmüller et al. [14], when assessing the conformance of a case in the BPIC2012 event log, around 41.3% of the information needs can be fulfilled with tables presenting case and/or event attributes and 34.8% by a process model. The remainder is often fulfilled with a line or bar chart, fulfilling 20.6% of the information needs. By taking into account the information needs of domain experts when visualising the trace, they are supported in their decision-making process and can better substantiate their key item selection.

3 Active Selection Approach

Taking into account the literature discussed above, we propose the Active Selection Approach. The goal of the approach is to structure the selection of key items using process data available during an audit. Fig. 1 gives an overview of the steps that make up the approach. The remainder of this section describes each step in more detail.

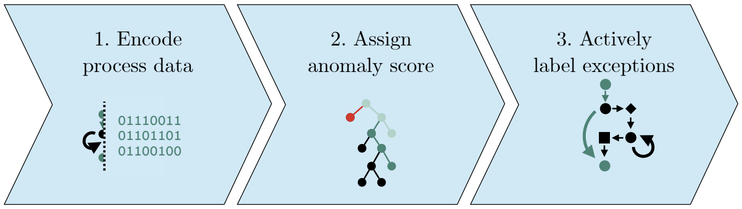


Fig. 1: Active Selection Approach

3.1 Step one: Encode process data

Before anomalies can be detected using traditional methods, the event log needs to be transformed into a tabular data structure where data is structured into rows, each of which contains information about a case, also known as *trace encoding* [15,16]. The way the traces are encoded should be tailored towards the process of the client and the type of information that should be retained. For example, if the domain experts are only interested in the resources and monetary value of the transactions, there is no need to consider the temporal aspect of the process data during encoding. Should some of the resulting features consistently have the same value as another feature or holds a constant value throughout the event log, they can be removed. Preferably, no further feature selection should be done, in order to retain as much information about the event log as possible.

3.2 Step two: Assign anomaly score

After the process data is encoded, an anomaly detection algorithm is used to assign an anomaly score to each case. The only constraint to the choice of algorithm is that it has to be an ensemble method so that the weight of the individual models can be updated based on domain expert feedback.

3.3 Step three: Actively label exceptions

With each case having an anomaly score, the cases with the highest anomaly score can be visualised and shown to a domain expert. Based on the visualisations, the domain expert has to label the case as either a key item or not. Based on the label, the weights of the algorithm are updated. This step has two benefits: 1) the domain experts gain insight into anomalous traces within the process of the client and 2) the weights of the algorithm are updated, potentially improving the results. After the domain expert has labeled a set number of cases, the algorithm assigns an updated anomaly score to each case. The result of the approach is an enriched event log containing updated anomaly scores. Based on these, the domain expert can decide to select certain cases with a high anomaly score as key items, thereby supporting their decision-making process.

4 Evaluation

The approach was implemented in Python (available on on Github[1]) and evaluated over three cycles with several domain experts: senior auditors from an audit firm, experienced students from a post-master accountancy program (around 2-4 years of practical experience), and attendees of a symposium on statistical auditing. Each cycle had two objectives: (1) evaluate the performance of the approach and (2) measure the saturation of the information needs of the domain experts with regards to the trace visualisation. During each cycle, domain experts completed a survey[2] that showed them six cases. Three of those cases were considered an exception (i.e. had a high anomaly score) and the other three were not (i.e. had a low anomaly score). The label given by the domain experts was viewed as the true label to later compute performance metrics. Table 1 provides an overview of who participated in each cycle and which sub-process they were shown.

4.1 Step one: Encode process data

The process data used during the evaluation is the publicly available Business Process Intelligence Challenge 2012 (BPIC2012) event log [17]. The event log contains 13.087 cases with 262.200 events. It describes an application process for a personal loan within a bank. The event log is chosen because the loan

[1] https://github.com/rubenpost/Model_agnostic_AAD/blob/main/main.py

[2] https://survey.uu.nl/jfe/form/SV_5jUGtcjPq1muHgG

Table 1: Evaluation cycles

Cycle	**Participants**	**Sub-process**	**# cases labeled**
One	3 senior auditors and 15 students	The offer (O)	108
Two	3 senior auditors and 53 students	The offer (O)	336
Three	3 senior auditors, 18 students, and 108 symposium attendees	The offer (O) and the work items (W)	648

applications contain financial information and could therefore realistically be part of an audit. There are 24 unique activities in the event log, representing three sub-processes. The event log contains three sub-processes, the application (A), the offer (O), and the work items (W) belonging to the application. To reduce the learning curve for domain experts when interpreting the information about the process, only one sub-process was used during each evaluation. This also reduced the number of activities the domain experts had to review. During the first and second cycle, the offer (O) was shown. In the third cycle, the offer (O) was shown to the auditors and students, while the work items (W) were shown to the professionals at the Limperg Symposium Statistical Auditing. Only accepted loan applications were included, as we assume that only these cases would have a financial impact on the client. The final event log contained 2.243 cases and 15.701 events.

The event log contained several attributes. In Table 2, the trace encoding used during this evaluation is described. All attributes were encoded as either aggregates or static. This means the order of the activities is lost, but the frequency is still kept as part of the feature. After trace encoding, the data had a shape of 2243x71, meaning there are 2.243 cases each represented by 71 features. Because of the limited moments available with the domain experts, the encoding type per attribute was not optimised based on the evaluation results.

Table 2: Feature encoding on BPIC2012 event log

Attribute	**Category**	**Type**	**Encoding**
CaseID	Case	Static	Not included
Resource	Event	Dynamic	Frequency
Activity	Event	Dynamic	Frequency
Timestamp	Event	Dynamic	Frequency
Registration	Event	Dynamic	Frequency
Status	Event	Dynamic	Frequency
Amount	Case	Static	As-is
Activity count	Case	Static	As-is
Case length	Case	Static	As-is

4.2 Step two: Assign anomaly score

During the evaluation, the Isolation Forest algorithm was used to assign an anomaly score to each case [18]. The Isolation Forest algorithm was used because it can cope with high-dimensional data sets, is generally fast, and is an ensemble method. The default parameters for the Isolation Forest as described in Das et al. [12] are used. Because of the way the approach is evaluated, the algorithm is not instantiated within the Active Anomaly Detection framework until the third cycle. For the first and second cycle, each case was assigned an anomaly score by Isolation Trees without individual weights. After assigning the anomaly score, the top and bottom 100 anomaly scores (viewed as exceptions and no exceptions, respectively) were used in the survey during cycle one and two. The labels received during the first and second cycle were used to update the weights of the algorithm for the third cycle.

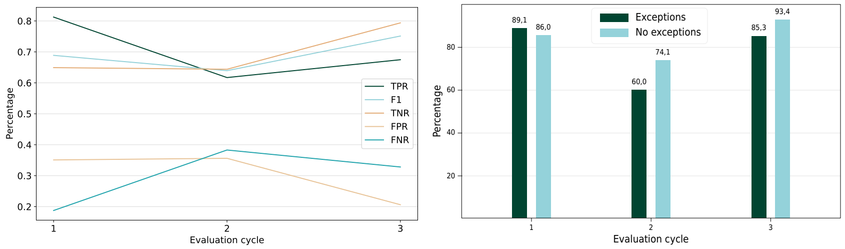
4.3 Step three: Actively label exceptions

The trace visualisation for the evaluation consisted of four different visuals. The first visual is a directly-follows-graph process model generated with PM4Py, a Python-based process mining package [19]. This type of graph is solely based on which activity directly follows which activity (i.e. a directly-follows dependency (a > b)) [20]. This means that concurrency and parallelism are ignored. This type of process model was chosen due to its ability to show the many loops a process can take [21]. In addition to the activities, the process model also includes the time between activities and time spent on the activity. One table visualises the directly-follows dependency between all activities in the case. The reason the directly-follows dependency was included in a separate table is that the frequency of the dependency is shown in the table, but not in the process model. Another table shows which resource performed which activities and how many times. Lastly, all numeric features of the case were plotted in a histogram. The bin in which the value of the case resides is highlighted.

4.4 Performance results

With the labels collected during the survey, the performance metrics were computed to evaluate the performance progression after each evaluation cycle. In addition to the performance metrics, the label confidence per label is computed, which shows how often domain experts agreed on the label of a case. The metrics are visualised in Fig. 2 (a). In this figure, the progression of the metric after each evaluation in a cycle is shown. The last result of the evaluation is the label confidence per evaluation cycle. In Fig. 2 (b), the label confidence is shown both for cases identified as exceptions and as not an exception by the approach.

After labeling the cases, the domain experts were asked two questions: *"What additional information would help in making your decision?"* and *"Did you have enough information to make your decision about each case?"*. These open questions relate to the second objective of the evaluation (i.e. are the information



(a) Performance metrics per cycle

(b) Label confidence per cycle

Fig. 2: Survey results

needs met?). The first question was an open question that aimed to provide feedback and points of interest for the approach. In addition to this open question, comments made by the domain experts were also written down and used as input for the next cycle. The answers to the second question also indicate the saturation of the domain experts' information needs and has three levels: *"yes"*, *"somewhat"*, and *"no"*.

The results of the first open question are shown in Table 3. The table describes which information needs were identified during each evaluation and how these were implemented in the next cycle (i.e. their impact) and Fig. 3 shows the indication of saturation per cycle.

Table 3: Suggested information needs and impact on approach

Cycle	**Suggested information needs**	**Impact**
One	Relation between number of resources used in the case and the norm	Included the average number of cancellations a case in the event log contains and how often a case with the same or more cancellations is found
	Relation between number of times a case is cancelled and the norm	Included the average number of resources a case in the event log contains and how often a case is performed by the same or more resources
	The directly follows relationship table is hard to interpret at first	Included an explanation of the table in the figure title
Two	Internal procedures for cancelling a case	None, this information is not available
Three	The financial impact of certain activities on the organisation	None, this information is not available
	Background of the process and the resources that work on it	None, this information is not available

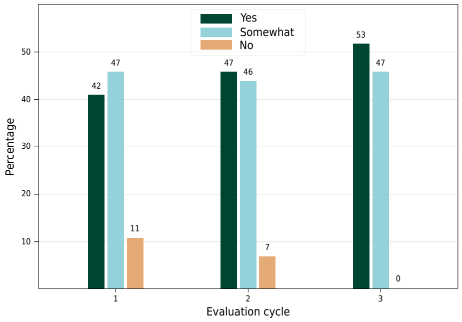


Fig. 3: Progression of information needs saturation per cycle

5 Discussion

In the next paragraphs, the results of the questions on information needs are discussed, the impact on the approach is described, and the performance metrics are interpreted per cycle.

5.1 Cycle one

During the first cycle, the domain experts seemed to agree that more context was needed on the case they were reviewing. Specifically, the attribute values of the case needed to be put in the context of the entire event log. Hence, the averages of several attributes deemed important by the domain expert are added to the trace visualisation. Besides this, they noted that the directly follows relationship table was hard to interpret. However, once they understood how to read the table, the information seemed very useful (mostly because it showed the order of the activities, something the model did not always do clearly).

The performance metrics show that the exceptions identified through the approach were quite often labeled as such by the domain experts. However, cases not identified as an exception were often labeled as an exception by domain experts as well. Hence, the FPR is slightly higher than the FNR. The F1 score was 68.9%, indicating that if the approach was used in a real audit, and the cases with the highest anomaly scores were selected as key items, it would select a key item most of the time. The label confidence was high in the first cycle (averaging 87.4%), meaning domain experts generally agreed on the label of a case. Hence, the robustness of the performance metrics of the first cycle is considered high. The performance metrics did not lead to further changes to the approach.

Based on these results, more context on the case was added to the trace visualisation. By adding the average over the entire event log and the rarity (i.e. how many other cases have the same values for that specific feature), the domain expert can compare the case to the ‘norm’.

5.2 Cycle Two

During cycle two, the information needs of the domain experts seemed to be more saturated. This can be seen both in the number of suggested information needs (only one) and the percentage of domain experts indicating they had enough information to review each case (93%). Except for the need for a more elaborate explanation of the histograms in the trace visualisation, the suggested information needs had no impact on the approach. This is because the information was not available during this research.

Despite the fact that the information saturation was higher in the second cycle, the performance metrics mostly decreased. The F1 score decreased to 63.4% (an 8% decline) and cases were more often predicted incorrectly. With the exception of the FNR, which increased with by 20%. This means domain experts were more likely to identify cases as an exception in general. The label confidence was also much lower, averaging 67.1%. The high information saturation and low label confidence indicate that professional judgement had a large influence on the performance metrics. This was not observed in the first cycle, but is further confirmed by domain experts indicating that working with process data is new and needs adjusting to, meaning that they have to rely more on their professional judgement than might be intended during an audit.

Based on these results, one minor change was made to the approach. An explanation was added to the histogram to describe what exactly the domain experts were looking at and the information the histogram gave them.

5.3 Cycle Three

During the last cycle, no further information needs were identified that had impact on the approach. Additionally, all of the domain experts indicated that they either had enough or somewhat enough information to review each case. None of the domain experts indicated they did not have enough information, indicating that the information needs were saturated the most in the third cycle.

The labels collected through the survey in the first and second cycle were used to update the weights of the algorithm. This led to an increase in performance metrics: the F1 score was the highest of all the cycles with 72.2%. A similar increase was also seen in the other performance metrics, meaning that domain experts were more likely to label a case the same way the approach did. The labeling confidence was also the highest out of all the cycles, averaging 89.4%. This cycle also evaluated two sub-processes. The increase in information saturation and performance metrics indicate that the approach generalised well to different sub-processes of the event log used during the evaluation. Because no further information needs were identified, no changes were made to the approach. This is in line with the measured information saturation.

6 Limitations

The study has potential limitations. The first limitation is the bias introduced by the domain experts when labeling the cases. This shows through the labeling confidence; the average labeling confidence is between 67.1% and 89.4%. This shows that labeling a case could involve a substantial amount of professional judgement. This could be related to the experience of the domain experts that participated in the survey, which was not always known. Besides the three auditors, the background and expertise of the students and professionals are unknown. This could lead to lower quality labels.

The approach requires domain experts to label cases before receiving a final list of identified exceptions. This might be cumbersome and could cause friction during the usage in audit. Regarding the results, it is unknown whether the results can be generalised to different event logs. Because only one event log, albeit divided into two sub-processes, is used during the evaluation, the results might not be reproducible with different event logs. The same is true for the encoding of the traces. Because different encoding types were not evaluated, it is unknown whether different trace encoding would improve the results.

7 Conclusion and Future Work

The evaluation showed that the approach has the potential to support the decision-making process of domain experts when selecting key items. Although auditors still need to make a manual selection of key items, they are able to better substantiate this selection. During the evaluation, multiple signs indicated that professional judgement had a large influence on the label domain experts gave a case (and therefore on the results). There were two reasons why professional judgement was still required. First of all, there was more uncertainty among the decision-makers with respect to the context of the process execution. During a normal audit, more contextual information is available. This could cause the domain experts to select more key items than they would normally do to reduce the risk of missing a misstatement. The second reason is that working with process data is new and needs adjusting to, meaning that domain experts relied more on their professional judgement then they normally would when selecting key items with less information about the behaviour of the transaction.

The subjectivity involved in labeling the cases should be further reduced. Currently, trace visualisation attempts to standardise the information on which the domain experts make their decision. By standardising this information, the decision of the domain experts becomes more structured and standardised, reducing the subjectivity involved during their decision-making. Future work should standardise the trace visualisation, further structuring the way professional judgement is used throughout the approach.

References

1. W. M. P. Van Der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Springer, 2016.
2. M. Jans, M. Alles, and M. Vasarhelyi, “The case for process mining in auditing: Sources of value added and areas of application,” *International Journal of Accounting Information Systems*, vol. 14, no. 1, pp. 1–20, 2013.

3. ——, "Process mining of event logs in auditing: Opportunities and challenges," *Available at SSRN: https://ssrn.com/abstract=1578912*, 2010.
4. International Standard on Auditing 450, "Evaluation of misstatement identified during the audit," 2009. [Online]. Available: https://www.ifac.org/system/files/downloads/a021-2010-iaasb-handbook-isa-450.pdf
5. W. C. Boynton and R. N. Johnson, *Modern auditing*, 8th ed. J. Wiley & Sons, 2001.
6. A. Sureka, "Kernel based sequential data anomaly detection in business process event logs," *arXiv preprint arXiv:1507.01168*, 2015.
7. K. Böhmer and S. Rinderle-Ma, "Multi-perspective anomaly detection in business process execution events," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2016, pp. 80–98.
8. T. Nolle, S. Luettgen, A. Seeliger, and M. Mühlhäuser, "Analyzing business process anomalies using autoencoders," *Machine Learning*, vol. 107, no. 11, pp. 1875–1893, 2018.
9. S. Pauwels and T. Calders, "An anomaly detection technique for business processes based on extended dynamic bayesian networks," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 494–501.
10. J. Ko and M. Comuzzi, "Detecting anomalies in business process event logs using statistical leverage," *Information Sciences*, vol. 549, pp. 53–67, 2021.
11. G. Schumann, F. Kruse, and J. Nonnenmacher, "A practice-oriented, control-flow-based anomaly detection approach for internal process audits," in *International Conference on Service-Oriented Computing*. Springer, 2020, pp. 533–543.
12. S. Das, W.-K. Wong, T. Dietterich, A. Fern, and A. Emmott, "Incorporating expert feedback into active anomaly discovery," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 853–858.
13. W. M. P. van der Aalst, M. de Leoni, and A. H. ter Hofstede, "Process mining and visual analytics: Breathing life into business process models," *BPM Center Report BPM-11-15, BPMcenter. org*, vol. 17, pp. 699–730, 2011.
14. C. Klinkmüller, R. Müller, and I. Weber, "Mining process mining practices: An exploratory characterization of information needs in process analytics," in *International Conference on Business Process Management*. Springer, 2019, pp. 322–337.
15. I. Teinemaa, M. Dumas, M. L. Rosa, and F. M. Maggi, "Outcome-oriented predictive process monitoring: Review and benchmark," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 2, pp. 1–57, 2019.
16. M. De Leoni, W. M. P. van der Aalst, and M. Dees, "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs," *Information Systems*, vol. 56, pp. 235–257, 2016.
17. B. F. van Dongen, "Bpi challenge 2012," 4 2012. [Online]. Available: https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204/1
18. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.
19. A. Berti, S. J. van Zelst, and W. M. P. van der Aalst, "Process mining for python (pm4py): bridging the gap between process-and data science," *International Conference on Process Mining*.
20. A. Augusto, R. Conforti, M. Dumas, and M. La Rosa, "Split miner: Discovering accurate and simple business process models from event logs," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 1–10.
21. W. M. P. van der Aalst, "A practitioner's guide to process mining: Limitations of the directly-follows graph," 2019.