Process Deviation Classification (Extended Abstract)

Manal Laghmouch
Faculty of Business Economics
UHasselt - Hasselt University
Hasselt, Belgium
manal.laghmouch@uhasselt.be

I. Introduction

Process mining is a family of process analysis techniques that enables the discovery of process models from an event log (process discovery), the ability to check conformance between the actual and the assumed process (conformance checking), and other process-related analyses to enhance the process (process enhancement) [1]. The process mining types process discovery and conformance checking are of primary interest in the context of auditing. While process discovery can be used to get an overall and objective understanding of the business processes of a company, conformance checking enables the automatic detection of process deviations between the actual and assumed process [2], [3].

Although conformance checking has great potential in detecting process deviations, challenges exist that hinder it from full adoption in auditing practice. First, because a normative process model is an idealistic and simplified representation of a process, performing a conformance check results in a large set of process deviations. This makes it impossible for the auditor to check each deviation one-by-one and therefore forces the auditor to take samples instead of auditing the complete set of transactions [4]. Second, the large amount of detected process deviations consists not only of deviations that are harmful from an auditing perspective (anomalies), but also of exceptional, but acceptable, behaviour (exceptions) [5]–[8]. Since the auditor is only interested in anomalies, the exceptions need to be filtered out. Manually classifying process deviations is a time consuming and costly task, making the use of conformance checking unpractical.

In this doctoral research, we aim to cope with the challenges associated with conformance checking in auditing by focusing on the classification of process deviations. For this purpose, we develop an active learning framework that combines machine learning techniques with domain expertise. The framework provides auditors with a practical approach to audit the complete set of transactions instead of a sample. The idea is to use conformance checking output (deviations) and combine this with domain expertise in an iterative process.

This extended abstract is structured as follows. In Section II, we describe our research methodology and walk through the different steps of our framework. An overview of related work is given in Section III. Finally, we conclude our research in Section IV.

II. Methodology

The doctoral thesis will consist of designing and testing a deviation classification framework. In this section, we present the current state of the framework and provide some information on how we will validate our study.

A. The framework

In our specific context, we are researching how we can make conformance checking feasible in auditing practice. To this end, we propose an active learning framework for the classification of process deviations. The proposed framework consists of the six following steps:

1) Label a small sample of the unlabeled set of deviating cases: The framework's goal is to label a set of deviating cases that were detected using conformance checking. The possible output labels for each deviating case are either *anomaly* or *exception*. An anomaly is a case that consists of at least one anomalous process deviation. An exception is a case that consists of only exceptional process deviations.

The framework starts with taking a small sample of a set of deviating cases that an auditor has to label as anomaly or exception. The labelled small sample is the input for the second step of the framework.

2) Mine rules from the labelled set: In the second step, we use the small labelled sample to mine rules. We want to discover two types of rules: control flow relations and other data relations. Control flow relations refer to relations between the order of the activities in a case. To discover such relations, we are planning to use a declarative constraint miner [9]. Other data relations can be discovered by using, for example, rule association mining [10]. Further research has to identify which algorithms best suit our context.

3) Transform rules to labeling functions: The mined rules from step 2 are in this step transformed into so-called labeling functions. labeling functions are a construction of rules used in the tool Snorkel. Snorkel is a system that implements weak supervision to train machine learning models without the need for labelled data. Instead, Snorkel uses labeling functions that encode domain knowledge in the form of rules [11]. labeling functions individually label deviating cases as anomalous or exceptional. This has as a consequence that they can overlap or conflict with each other. For that reason, labeling functions are combined in a generative model to predict a final (unambiguous) label [12]. We choose to use

Snorkel because it can indirectly use domain knowledge to guide the labeling of individual cases.

4) Combine labeling functions in a machine learning model and label the remaining unlabeled cases: In this step, we combine the labeling functions of Snorkel in a machine learning model. Snorkel has a built-in generative model, called the LabelModel. The LabelModel weights and calculates labeling function accuracy to assign a final weighted label to each case in the data [11]. The result of this step is a set of cases that are labelled as anomaly or exception.

5) Calculate the accuracy: In the last step of the framework, we calculate the model accuracy by using a labelled validation set. If the accuracy is below a threshold (that is beforehand defined by the auditor), then we re-iterate over the framework by taking an additional sample of the unlabeled data. If the accuracy is above the predefined threshold, then the framework stops here, and the auditor ends up with the set of labelled cases. The auditor can now focus the audit on anomalous cases.

B. Validation

The results of this doctoral study will be validated in two steps. First, we artificially create event logs and process models on which we apply our framework. Our approach will be validated based on the experimental results. We plan to look at the model accuracy, the impact of high or low-quality rules on model performance, the impact of the number of rules on model performance, and the number of iterations over the framework.

Second, after succeeding in the artificial setup, we apply the framework in a real-life setting. For this purpose, we are working with a Big Four auditing firm. In this stage, we will again look at model performance, the impact of the quality and quantity of given rules, and the number of iterations over the framework.

III. Related work

Since data is widely available and technology is becoming better, auditing is forced to change [7]. The continuous auditing field of research response to such changes by automating auditing procedures [5], [13]. A family of techniques that has got increased attention lately is process mining. With only an event log as input, process mining can provide auditors with objective views on the process [1], [14]. For in-depth analyses, the type conformance checking is relevant. However, it detects an alarm flood of deviations between the assumed and real process of a company that is impossible to check manually. Consequently, conformance checking does not support a full-population audit yet [5].

Although previous research proposes frameworks to cope with alarm floods in continuous auditing [5], [15], [16], a full practical elaboration misses. This research proposes a practical deviation classification framework that enables the automatic labeling of process deviations in auditing.

IV. Conclusion

In this doctoral research, we want to enable full population tests in auditing by making conformance checking more feasible in practice. We propose an active learning framework that combines conformance checking output with domain knowledge with the goal to label deviating cases as anomaly or exception. The research follows a two-step approach. First, we set up some experiments on artificially generated event logs and process models. After that, we test the framework on a real audit environment of a Big Four auditing firm.

References

[1] W. Van der Aalst, *Process mining: Data Science in Action*, 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] R. J. C. Bose and W. van der Aalst, "Trace alignment in process mining: opportunities for process diagnostics," in *International Conference on Business Process Management*. Springer, 2010, pp. 227–242.

[4] M. G. Alles, A. Kogan, and M. A. Vasarhelyi, "Putting continuous auditing theory into practice: Lessons from two pilot implementations," *Journal of Information Systems*, vol. 22, no. 2, pp. 195–214, 2008.

[5] M. Jans and M. Hosseinpour, "How active learning and process mining can act as Continuous Auditing catalyst," *International Journal of Accounting Information Systems*, vol. 32, pp. 44–58, Mar. 2019.

[6] B. Depaire, J. Swinnen, M. Jans, and K. Vanhoof, "A process deviation analysis framework," in *International Conference on Business Process Management*. Springer, 2012, pp. 701–706.

[7] D. Y. Chan and M. A. Vasarhelyi, "Innovation and practice of continuous auditing," *International Journal of Accounting Information Systems*, vol. 12, no. 2, pp. 152–160, 2011.

[8] M. A. Vasarhelyi, M. G. Alles, and A. Kogan, "Principles of analytic monitoring for continuous assurance," *Journal of emerging technologies in accounting*, vol. 1, no. 1, pp. 1–21, 2004.

[9] M. Pesic, H. Schonenberg, and W. M. Van der Aalst, "Declare: Full support for loosely-structured processes," in *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*. IEEE, 2007, pp. 287–287.

[10] C. Zhang and S. Zhang, *Association rule mining: models and algorithms*. Springer, 2003, vol. 2307.

[11] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: rapid training data creation with weak supervision," *The VLDB Journal*, vol. 29, no. 2, pp. 709–730, 2020.

[12] A. J. Ratner, S. H. Bach, H. R. Ehrenberg, and C. Ré, "Snorkel: Fast training set generation for information extraction," in *Proceedings of the 2017 ACM international conference on management of data*, 2017, pp. 1683–1686.

[13] M. A. Vasarhelyi and F. B. Halper, "The continuous audit of online systems," in *Auditing: A Journal of Practice and Theory*, 1991.

[14] 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.

[15] J. L. Perols and U. S. Murthy, "Information fusion in continuous assurance," *Journal of Information Systems*, vol. 26, no. 2, pp. 35–52, 2012.

[16] P. Li, D. Y. Chan, and A. Kogan, "Exception prioritization in the continuous auditing environment: A framework and experimental evaluation," *Journal of Information Systems*, vol. 30, no. 2, pp. 135–157, 2016.