Towards Full Population Testing in Auditing: How Many Process Deviations Should Be Labeled?

DOI: 10.1109/ICPM63005.2024.10680672
Handle: http://hdl.handle.net/1942/44827

Towards Full Population Testing in Auditing: How Many Process Deviations Should Be Labeled?

Manal Laghmouch
EDM
Hasselt University
Maastricht University
Belgium - The Netherlands
0000-0002-6513-2587

Benoît Depaire
EDM
Hasselt University
Hasselt, Belgium
0000-0003-4735-0609

Mieke Jans
EDM
Hasselt University
Maastricht University
Belgium - The Netherlands
0000-0002-9171-2403

***Abstract*—Conformance checking allows auditors to detect process deviations automatically, resulting in numerous deviations, with only a few being relevant. Identifying notable items amidst this large data set is challenging. Machine learning techniques offer potential solutions, but questions about the required number of labeled deviations and the impact of label quality remain. Our study investigates these factors' effects on Decision Trees and Random Forests. Results demonstrate these models' effectiveness in identifying notable items within imbalanced deviation populations. Achieving 90% precision and recall is feasible with about 400 to 600 labeled deviations, depending on the notable items' population fraction. A higher fraction of notables reduces the required labeled deviations. Varying label quality produced similar results. Additionally, classifications identifying at least 90% notable items are linked to less complex processes.**

***Index Terms*—Auditing, Conformance Checking, Deviation Classification, Machine Learning, Notable Item, Process Deviation, Process Mining**

I. Introduction

Auditors are responsible for investigating a company's financial reports and disclosures and assuring that the statements truly represent the company's financial state. To achieve this, auditors delve into the company's business environment, often gaining insights through examining its business processes. Analyzing business processes offers numerous benefits to the audit [1]. Employing automated process analysis techniques in auditing allows for a detailed examination of the company's adherence to established procedures. In particular, conformance checking is used to detect mismatches between recorded transactions and a normative process model. The result is a comprehensive list of process deviations [2].

The main advantage of data-driven process analysis in an audit context lies in its ability to screen the entire set of business transactions [3]. Although this bears the potential to identify all deviating transactions without the need for upfront sampling in auditing, a challenge arises in managing a large number of detected deviating transactions in the subsequent audit steps. It is known that the set of detected process deviations comprises a large set of (justifiable) exceptions and a small set of notable items. Only the notable items are deviations that are relevant to the auditor. In general, notable items only make up for about 5% of the total number of deviations [4].

Identifying notable items within the pool of process deviations is a complex task that, if done manually, becomes practically impossible [5]. Moreover, human limitations in processing vast amounts of information, particularly in accounting, further complicate the situation [6]. Case studies have demonstrated that auditors can be overwhelmed by information overload resulting from a substantial set of process deviations [7]. Therefore, the main challenge is to find a practical approach for processing the identified deviations while maintaining the advantage of including the full population of transactions in the audit.

Machine learning techniques can be used to address this challenge by automating the identification of notable items within the full set of detected deviations [8]. Despite attempts to tackle this challenge through clustering, prioritization, and machine learning techniques (e.g., [5], [9], [10]), in the end, these methods still largely hinge on sampling approaches. This prevents auditors from thoroughly assessing the entire population of deviating transactions. It is important to note that existing methods are predominantly of a conceptual nature with limited empirical evidence to support the theoretical claims. Consequently, several questions remain unanswered. How many labeled deviations are required for such frameworks to work? What about the quality of the provided labels; to what extent does this affect a model's performance?

If one wants to pursue full-population testing in auditing, a labeled set of process deviations is a general requirement to train a machine learning model. However, if a substantial amount of data needs to be manually labeled, the economic feasibility of this approach comes into question. In extreme cases, the requirement for extensive manual labeling could negate the advantages of applying machine learning to classify process deviations.

To bring full-population-based auditing closer to reality, it is, therefore, essential to understand the required number of labeled process deviations to train a classifier. This paper provides this understanding. The contribution of this paper is as follows:

1) We demonstrate that achieving precision and recall of 90% is feasible with about 400 to 600 labeled deviations. The required quantity depends on the fraction of notable

items present within the respective full population of deviations.
2) We find that both DTs and RFs exhibit comparable precision, but RFs outperform DTs in recall.
3) We find that varying label quality between 75 and 95% does not affect performance.
4) Classifications identifying at least 90% notable items appear to be associated with less complex processes compared to those identifying fewer notable items.

The remainder of this paper is structured as follows: Sect. II outlines and explains the experimental design. Sect. III shows the study results. Sect. IV discusses the results. Sect. VI provides some process mining work related to this study. Sect. VII concludes the paper.

II. Experimental Design

This section outlines the experimental design objectives, the process of generating synthetic data, and the training of the machine learning classifier. Codes and data are available on GitHub[1].

A. Research Goal and Design

The primary goal of our study is to determine how many process deviations should be labeled as exceptions or notable items to achieve a specific level of performance. To this end, we create a synthetic dataset of process deviations that simulate an audit engagement setting. These deviations represent cases where the process execution (i.e., trace) does not align with the desired business process model. consequently, this study focuses on control-flow deviations.

These deviations are categorized as either exceptions or notable items. Exceptions are deviations from the desired process model but align with a more loosely defined variant of it. Notable items, on the other hand, are deviations that do not conform to either the desired process model or its loosely defined variant. We use DeclareMoGeS [11] to generate this setup.

The generated data is the input to different classifiers. The goal of the classifier is to distinguish between notable items and exceptions. Our primary focus is on identifying notable items (i.e. the minority class). The performance of these classifiers is measured by precision and recall metrics because they are the most meaningful in an auditing context. Precision is expressed as the percentage of classified notable items that are true notable items. Recall reflects the percentage of true notable items identified as notable item by the classifier. We put the performance threshold at 90% for precision and recall.

To gain insights into how many labeled deviations are required to classify an imbalanced set of process deviations, we perform a series of computational tests. Particularly, we want to measure whether the following parameters have an impact on the model's performance:

- the training set size;
- the quality of the provided labels in the training set.

[1]https://github.com/manallaghmouch/FullPopulationAudit

The training set size is the number of labeled deviations needed to train the classifier. To control for the quality of the provided labels, we add noise to the generated training sets. The percentage of noise added is a proxy of how accurate the auditor is at providing correct labels. We add 5% label noise in our initial setup.

Furthermore, we will compare classifications not reaching the performance threshold of 90% to classifications reaching this threshold with regard to some context variables, such as the complexity of the underlying process model from which the deviations are discovered. The complexity of the process model is expressed as the number of constraints and activities in the model, as we use declarative process models to start from.

B. Data Generation

The data used to train the classifier was synthetically generated. We opted for synthetic data because this allows to unveil causal patterns between features in the dataset [12]. Figure 1 shows a visual representation of the synthetic data generation process after model generation. The steps followed to achieve the final training sets are explained in the following paragraphs.

1) Generating Synthetic Process Models: To generate training data on deviations, we need process models from which to deviate. Declarative process models were selected to start from, given their flexibility in describing business processes and their alignment with the auditing context. Auditors often conceptualize processes in terms of rules.

Declarative process models consist of constraints that define the boundaries within which a process has to be executed [13]. For example, a constraint might be *"Response(create order, approve order)"*, meaning that each created order should eventually be followed by an approval within one process execution. If an execution complies with the constraint, then the constraint is *satisfied*. If not, the constraint is *violated*. A process execution fits a model if and only if it satisfies all constraints in the model.

As a first step, sets of declarative process models were generated. Each generated set of models includes three models that are hierarchically related to each other in terms of allowed process behaviour:

- The *world model* represents how the process actually operates in reality, encompassing all behaviors, including those outside the auditor and normative models explained below.
- The *auditor model* is more restrictive and focuses on identifying notable items, which are deviations considered genuinely incorrect. This model often includes the auditor's implicit knowledge, which is not documented.
- The *normative model* is the most restrictive, identifying all deviations, including both notable items and exceptions. This model typically comprises explicit, documented knowledge and is in practice used for conformance checking.

Fig. 1: Visual representation of data generation process after model generation

We generated 100 model sets of hierarchical process models using the DECLARE Model Generator and Specializer DeclareMoGeS [11]. First, 100 normative models are generated. These models comprised 10 to 26 constraints and 10 to 26 activities. We define our initial models as broadly as possible to ensure that the findings of our study can be extrapolated to settings that meet our specifications.

For each normative model, an associated auditor model and a world model are generated. Each successive model is less restrictive than the previous one: the auditor model allows for more behaviors than the normative model, and the world model further loosens behaviors allowed by the auditor model. All models within the same set contain the same number of activities, but the constraints in each more restrictive model specifically limit the behaviors permitted by the less restrictive models [14].

In the less restrictive model, any constraints that could be relaxed were relaxed. For instance, consider a normative model with 10 constraints. One of these constraints is *"Chain Response(create order, approve order)´´*, which means that if an order is created, then *immediately* afterward, it should be approved. This constraint can be relaxed in the auditor model to *"Response(create order, approve order)´´* imposing that the creation of an order should *eventually* be followed by an approval. This process of relaxing constraints is applied to all constraints in the normative model. If a less restrictive version of a constraint does not exist, the original constraint is retained.

2) Constructing Imbalanced Full Population: We generated synthetic event logs with Declare4Py [15], employing the models that were generated in the previous step. To generate the logs, the least restrictive model from each model set was used, the world model. Each world model resulted in an event log consisting of 10 000 traces, each containing 8 to 15 events. We select these specifications to be as realistic and comprehensive as possible, ensuring that the findings of our study can be applied to similar contexts.

Subsequently, the event logs were subjected to two conformance checks:

1) First Conformance Check: Comparing the event log against the normative model to identify *process deviations*. Here, process deviations are defined as transactions that do not comply with the normative model. A transaction consisting of multiple events is considered as a whole; thus, deviations are assessed at the case level, not the event level.
2) Second Conformance Check: Comparing the set of deviating traces against the auditor model to distinguish between exceptions (conform with the auditor model) and notable items (not conform with the auditor model).

This process provided 100 full populations of labeled process deviations. To simulate the real-world auditing environment, where a large set of deviations contains only a small percentage of notable items, we adjusted the data to create imbalanced datasets. We applied three different percentages of notable items in the data (1%, 2%, and 5%) to each full population, theoretically resulting in 300 adjusted (imbalanced) datasets (3×100). However, it was not possible to construct the desired imbalance for every population. For example, if the number of notable items or exceptions was initially zero, achieving the desired imbalance was impossible. In such cases, the dataset was excluded from the experiments. In total, 18 datasets were excluded for this reason. Therefore, this step resulted in 282 imbalanced full populations of deviating transactions.

3) Constructing Training Set with Noise: From each imbalanced population, we draw a maximum of 100 samples (282×100). A sample was constructed by picking random instances from the full population given a certain sample size. The sample sizes varied uniformly between 100 and 1000 deviations. In some cases, it was impossible to draw the requested sample because the requested size was larger than the population size. This step resulted in 24 163 samples, which we refer to as our *labeled (deviation) samples*.

Each deviating transaction in the labeled sample was transformed into a format suitable for classification. Each constraint from the normative model was included as a boolean feature

indicating whether the constraint was violated (1) or not (0). The labels ('notable item' or 'exception') were included for training purposes.

To ensure realism, we introduced *label noise* to the labeled sample. Label noise refers to the situation where the labels in the sample are not the ground truth [16]. A small percentage of labels was flipped to simulate potential misclassifications by auditors, meaning that some transactions labeled as notable items were actually exceptions and vice versa. We inject 5% label noise into the labeled sample. This translates to an auditor who is 95% accurate in providing correct labels. The result of this step is 24 163 noisy training sets.

C. Training Machine Learning Model

Given the imbalance in the 24 163 final training sets, we used Random Oversampling to balance each set before classification. This resulted in balanced training sets, with each set consisting of an equal number of notable items and exceptions. Subsequently, all training sets were used to train a classifier. We assessed the classifier's performance on the remaining set of deviations from the imbalanced population from which the training set was drawn. We measured performance using precision and recall.

DTs and RFs were chosen as classifiers. DTs were selected for their interpretability, which is crucial in auditing to ensure the classifications made by the classifier are understandable. However, since they may be more sensitive to training set variability, we also assessed performance on RFs. RFs are expected to reduce sensitivity to training data variability. Therefore, we expect RFs to provide more stable and reliable performance metrics than DTs.

III. Results

In this section, we present our experimental findings. We investigated how the quantity and quality of labeled process deviations impact the performance of DTs and RFs. Subsequently, we compare high-performance to low-performance classifications.

A. Effect of Training Set Size on Performance

As described in the previous section, we created 282 populations of process deviations. These populations averaged 5451 deviations, with the smallest consisting of 160 deviations and the largest consisting of 9963 deviations. From each population, 100 labeled deviation samples were randomly drawn to serve as training sets. The size of the labeled samples varied between 100 and 1000 deviations, with an average of 540 deviations. Altogether, 24 163 training sets were used for classification. Each training set was used once to classify process deviations from its respective population, ensuring the independence of each run.

For each of the 24 163 training sets, the performance of a DT and RF classifier was measured in terms of precision and recall. Every classification was repeated ten times to obtain an average result for the metrics. We investigated the effect of the training set size on the classifier's precision in identifying notable items for different levels of notable item prevalence in the population. We define 90% as the threshold for satisfactory performance in an auditing setting.

Figure 2 visualizes our findings for DTs (subfigures (a) and (c)) and RFs (subfigures (b) and (d)). Each bar in the figures represents the average performance the classifier achieves for a given training set size (see legend) and a given fraction of notable items present in the population (see X-axis). By considering the fraction of notable items present in the population, we can provide a nuanced view of the number of labeled deviations needed.

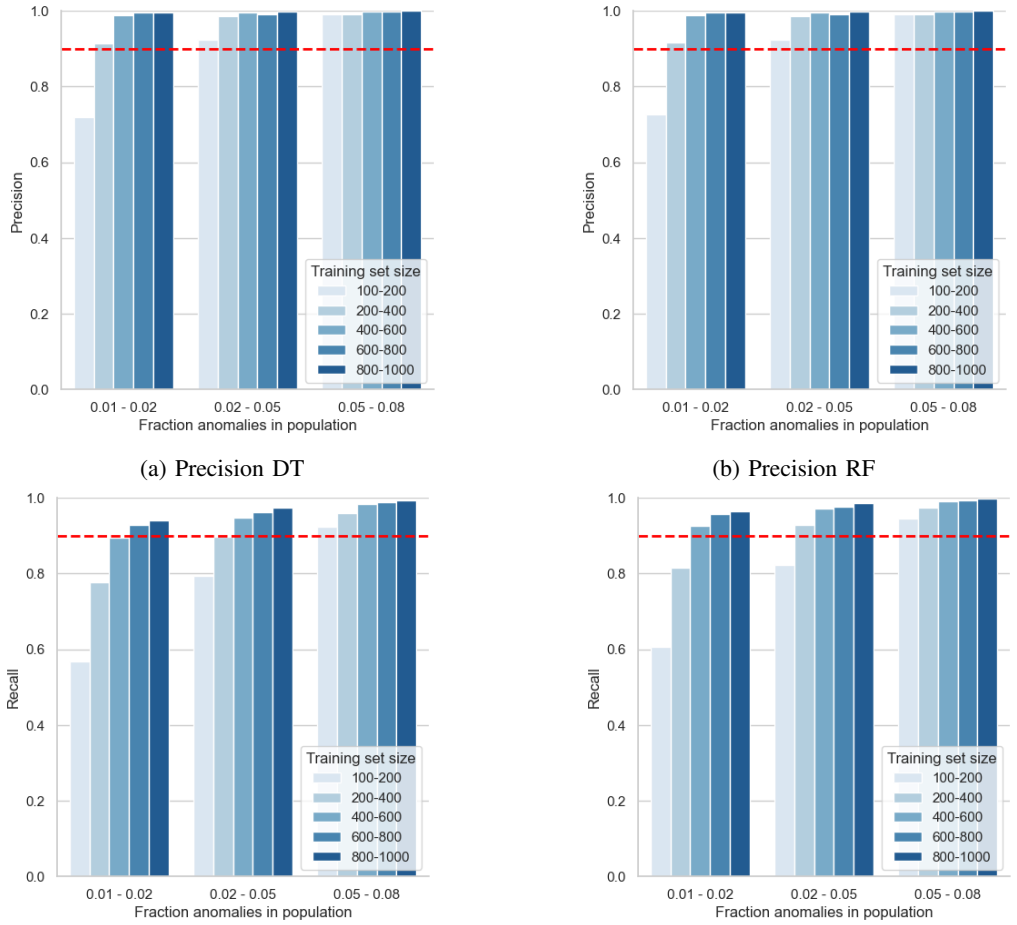
Our findings show that the precision generally increases with the increase in training set size. Smaller training set sizes consisting of 100 to 200 deviations exhibit lower precision, especially if only 1 to 2% notable items are present in the population from which the training set was drawn. As the training set size increases, the precision stabilizes and remains high across different fractions of notable items in the population of process deviations (Figure 2 (a) and (b)). Both the DT and RF models reach a precision of 90% for a training set consisting of at least 200 labeled deviations. Most classifications resulted in a precision near 1, meaning that if the model predicts a deviation as anomalous, it is almost always correct.

Recall also improves with the increase in training set size. For very low fractions of notable items present in the population (1 to 2%), smaller training set sizes (100 to 400 deviations) result in a recall below the threshold of 90%. Larger training sets consisting of more than 600 labeled deviations achieve high recall across all notable item fractions in the population (Figure 2 (c) and (d)).

Some differences are noticeable between the recall of the DT and RF model. For DTs, more data has to be labeled to reach a recall of at least 90% (at least 400 deviations for all notable item fractions), while with RFs less effort is required. For deviation populations comprising 1 to 2% notable items, this translates to at least 400 deviations. For populations with 2 to 5% notable items, this is at least 200 deviations. For populations with 5 to 8% notable items, a set size of only 100 to 200 deviations suffices.

Figure 3 visualizes the difference in variation between the DT and RF models. The variation is smaller for the RF classifier than it is for the DT classifier. Since recall reflects the percentage of true notable items identified as notable items by the classifier, auditors might prefer a higher value for this metric. Therefore, RFs are preferred in this context.

Overall, an increased number of training samples improves both precision and recall. This suggests that the DT and RF model benefit from having more data to learn from, leading to better predictions. However, it is noticeable that when a larger percentage of notable items is present in the process data, fewer data need to be labeled to identify the notable items. In other words, the auditor has to put in less effort to label deviations if the process results in more notable items. Conversely, more streamlined processes with fewer notable items require more attention because the auditor has to label



(a) Precision DT

(b) Precision RF

(c) Recall DT

(d) Recall RF

Fig. 2: Performance classifier for different fractions of notable items in training set

more data to ensure a reasonable number of notable items are identified.

B. Effect of Label Quality on Performance

To test the effect of label quality on performance, we constructed additional experimental setups. Besides the initial setup in which 5% label noise was injected into the training set, we constructed new setups with 10, 15, and 25% label noise, each decreasing the quality of the provided labels. Each training set in each setup is constructed as explained in Section II, but then for the respective noise level.

By changing the label noise, we want to test to what extent auditors who provide less accurate labels affect the performance of DT and RF models Figure 4 shows our findings.

The figures show that precision and recall stay stable across different noise levels. We also conducted this analysis for different percentages of notable items present in the training set, but the observation stayed the same. Furthermore, the figures also confirm our previous finding that after labeling about 400 process deviations, the performance of the machine-learning models stabilized in terms of precision. For recall, this stabilization occurs later.

C. High- versus Low-Performance Classifications

Context variable	Classifier	t-statistic	p-value
Fraction notable items in train. set	DT	28.312	<0.01
Fraction notable items in train. set	RF	28.167	<0.01
# process activities	DT	-0.568	0.570
# process activities	RF	-0.536	0.592
# constraints normative model	DT	-1.832	0.067
# constraints normative model	RF	-1.576	0.115
# constraints auditor model	DT	-1.059	0.290
# constraints auditor model	RF	-0.796	0.426

Table 1: Results of t-tests comparing high-precision ($\geq$ 0.9) to low-precision ($<$ 0.9) classifications

To gain insight into what distinguishes high-performance classifications from those reaching a lower performance, we compare classifications reaching the performance threshold of 90% to classifications not reaching the threshold. The comparison is based on some contextual parameters, such as the fraction notable items in the training set, the number of

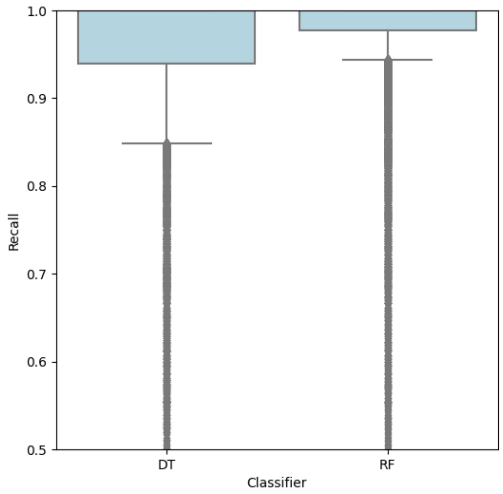


Fig. 3: Variability in recall of Decision Trees (DT) and Random Forests (RF)

Context variable	Classifier	t-statistic	p-value
Fraction notable items in train. set	DT	47.516	<0.01
Fraction notable items in train. set	RF	47.516	<0.01
# process activities	DT	-11.306	<0.01
# process activities	RF	-15.732	<0.01
# constraints normative model	DT	-24.701	<0.01
# constraints normative model	RF	-19.259	<0.01
# constraints auditor model	DT	-28.260	<0.01
# constraints auditor model	RF	-19.801	<0.01

Table 2: Results of t-tests comparing high-recall ($\geq$ 0.9) to low-recall ($<$ 0.9) classifications

activities occurring in the underlying process, the number of constraints present in the normative process model, and the number of constraints present in the auditor process model.

We define high-precision (low-precision) classifications as classifications for which the DT and RF models reached at least (did not reach) 90% precision. Similar definitions are applied for high-recall and low-recall classifications. We conducted t-tests to reveal differences among the groups. The findings of our analyses are shown in Tables 1 and 2 for, respectively, precision and recall.

Regarding high-precision and low-precision classifications, no significant differences are visible in terms of the number of activities present in the underlying process, the number of constraints in the normative model, and the number of constraints in the auditor model. However, high-precision classifications significantly differ from low-precision classifications in the fraction of notable items present in the training set used to train the classifier ($p < 0.01$). A higher precision seems associated with training sets with more notable items. These findings apply to classifications made by both the DT and RF models.

Regarding recall, more differences are visible. Similarly to precision, high-recall classifications significantly differ from low-recall classifications in the fraction of notable items present in the training dataset ($p < 0.01$). Furthermore, high-recall classifications significantly differ from low-recall classifications in terms of the number of activities occurring in the underlying process ($p < 0.01$), the number of constraints in the normative model ($p < 0.01$), the number of constraints occurring in the auditor model ($p < 0.01$). High-recall classifications concern processes with fewer activities, fewer constraints in the normative model, and fewer constraints in the auditor model than low-recall classifications. Since the number of activities in a process and the number of constraints in a process model indicate how complex a process is, we could state that both for the DTs and RFs, a higher recall is associated with less complex processes than it does for more complex processes.

IV. Discussion

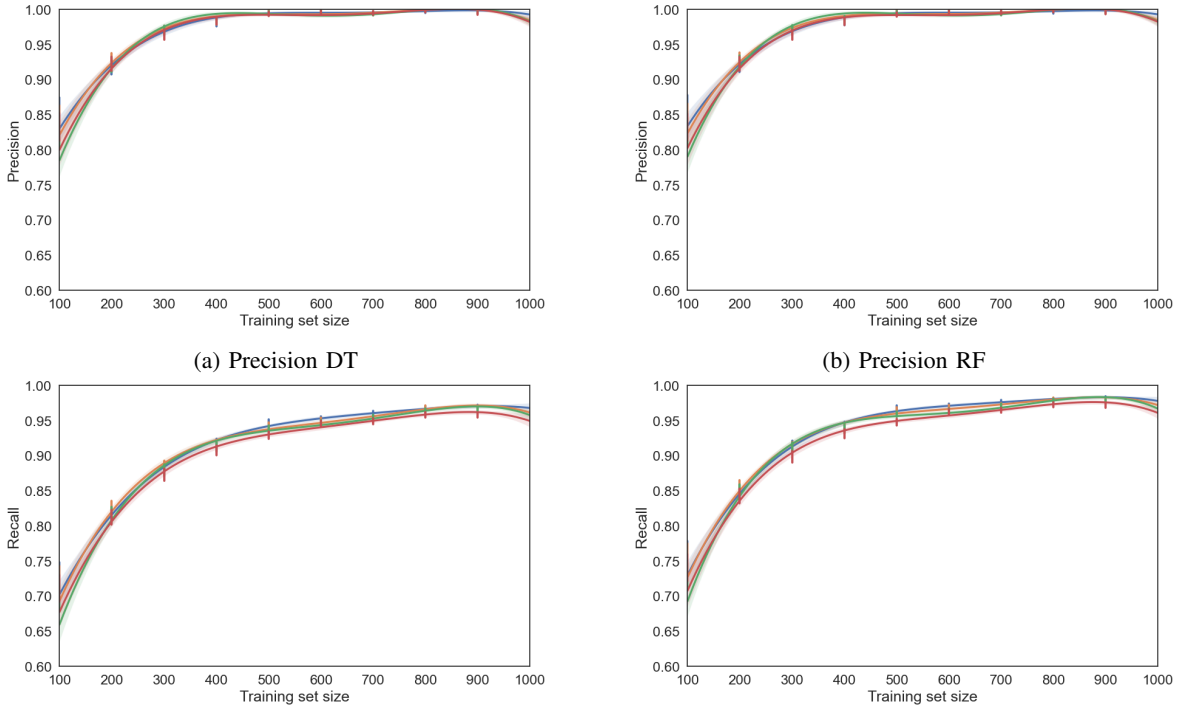
The findings of our study are particularly of interest to auditors. Continuous Auditing research pleads to automate some audit tasks such that the auditor can focus on tasks for which human expertise is of substantial value [17]. One of the proposed automations concerns the identification of process deviations, which can be accomplished by applying conformance checking. Despite good intentions, this automation does not really result in the desired effect of the auditor focusing on expert tasks. Conformance checking results in a set of process deviations that is too large to process manually [4].

Frameworks have been proposed to solve the challenge introduced by conformance checking output, but they have not found a way to practice because some questions remain unanswered. One of the most prevalent questions concerns the number of labeled deviations needed to reach satisfactory performance of a machine learning model to classify all deviating transactions.

Our study provides a nuanced answer to this question. Firstly, the number of labeled deviations required to achieve satisfactory performance depends on the proportion of notable items within the population. When notable items are more prevalent, fewer labeled samples are needed to achieve high precision and recall. This implies that in scenarios where notable items are expected to be more frequent, auditors can effectively use machine learning models with relatively less labeled data to maintain certain performance.

This finding appears contradictory. In practice, an auditor is more satisfied when a process concerns fewer notable items, and a higher assessed risk would usually invoke more testing to provide assurance over the financial reports [18], [19]. However, our study indicates that a higher prevalence of notable items in the population of deviations reduces the auditor's manual work as less manual labeling is needed to create a proper training set for the classifier.

Secondly, our experiments showed a similar average recall for Decision Trees and Random Forests. However, the recall of Random Forests varied less, meaning that the percentage of true notable items identified as notable items was about the same across different classifications. In an auditing context,



(a) Precision DT

(b) Precision RF

(c) Recall DT

(d) Recall RF

Fig. 4: Performance of Decision Trees (DT) and Random Forests (RF) for different noise levels in training set (5% (blue), 10% (orange), 15% (green), 25% (red))

it is preferred that the ability to identify true notable items remains consistent across individual audits.

Furthermore, classifications identifying more than 90% of the notable items seem associated with less complex processes than classifications identifying less than 90% notable items. Simplified processes often coincide with robust internal control systems. These controls act as checks and balances, minimizing the occurrence of notable items or errors [19]. Consequently, when notable items do occur, they might be more conspicuous and, therefore, easier to detect.

V. Future Research

While this study demonstrates the feasibility of full population testing in a synthetic audit environment, several research avenues remain. Our first next step is to replicate this study on real-life auditing data for validation purposes, and to expand the notion of process deviations to other perspectives (beyond control-flow).

Second, while Decision Trees and Random Forests achieved over 90% recall and precision, advanced methods like deep learning might offer better performance. However, the trade-off between performance and explainability needs consideration.

Lastly, examining the relative importance of notable items is crucial, as their impact on financial statements or audit priorities can vary. Developing methods to assess and incorporate these differences could improve classifications.

VI. Related Work

In this section, we contextualize our research based on some related work.

A. Approaches to Handle Overload of Deviations

Handling the overload of detected deviations can be approached by drawing high-quality samples [20]–[23], clustering or prioritizing [5], [9], [10], or by conducting full population tests [8], [24]. Our work distinguishes itself by addressing a complementary problem: determining the number of labeled deviations required to train a supervised machine learning model to achieve satisfactory performance in auditing. Rather than comparing our method to others, we aim to offer guidelines on the minimal labeling effort required for full population testing.

Unsupervised outlier detection methods [25] provide an alternative approach to supervised classification in auditing. They identify unusual patterns without labeled data, which is useful in initial analyses. However, they may not capture auditing-specific nuances. Supervised techniques leverage auditors' domain knowledge in labeled data for more precise, contextually relevant distinctions, essential for accurate deviation interpretation [26].

B. Hierarchy of Declarative Process Models

As part of our research design, we generated sets of hierarchical process models to mimic an auditing context. More specifically, we created DECLARE process models. DECLARE

is a declarative process language that describes a process by a set of constraints or business rules. In contrast to procedural process models, a DECLARE model allows for all process behaviour that is not explicitly forbidden by the constraints in the model. Since declarative process models are a set of rules [13], This aligns well with how auditors conceptualize processes.

The concept of hierarchy in process models has been explored in previous research [14], [27], [28]. Process model hierarchies were crucial for our study, as they allowed us to create models that progressively relax or tighten constraints, mimicking the auditors' challenge of identifying relevant notable items.

VII. Conclusion

Conformance checking provides auditors with a technique to automatically detect process deviations but bears a challenge: the set of detected deviations is too large to process. Machine learning could tackle this challenge. However, some questions remain unanswered, hindering their practical application. How many process deviations should be labeled to reach a certain performance? And what about the quality of the provided labels? This paper answers these questions by training DT and RF models on synthetic yet realistic data.

We demonstrate that a precision and recall of 90% can be attained with a limited number of labeled deviations. However, the required quantity hinges on the proportion of notable items within the detected process deviations population. RF models emerged superior in achieving high recall due to their more consistent performance. While the average recall of DTs and RFs was similar, RFs are preferable in auditing contexts since not only the average recall across all audits is of interest but also the recall of each individual audit. Furthermore, classifications identifying at least 90% notable items seem to be associated with less complex processes compared to those identifying fewer notable items.

Acknowledgments

Manal Laghmouch thanks Research Foundation Flanders for the SB PhD fellowship (1S40622N) supporting this research.

References

[1] M. Werner, M. Wiese, and A. Maas, "Embedding process mining into financial statement audits," *International Journal of Accounting Information Systems*, vol. 41, p. 100514, 2021.

[2] M. Werner, "Financial process mining-Accounting data structure dependent control flow inference," *International Journal of Accounting Information Systems*, vol. 25, pp. 57–80, 2017.

[3] S. Groomer and U. Murthy, "Continuous auditing of database accounting systems using embedded audit modules," *Journal of Information Systems*, vol. 3, no. 1, pp. 53–69, 1989.

[4] M. Jans, M. G. Alles, and M. A. Vasarhelyi, "A field study on the use of process mining of event logs as an analytical procedure in auditing," *The Accounting Review*, vol. 89, no. 5, pp. 1751–1773, 2014.

[5] T. Chiu and M. Jans, "Process Mining of Event Logs: A Case Study Evaluating Internal Control Effectiveness," *Accounting Horizons*, vol. 33, no. 3, pp. 141–156, Jun. 2019.

[6] E. R. Iselin, "The effects of information load and information diversity on decision quality in a structured decision task," *Accounting, organizations and Society*, vol. 13, no. 2, pp. 147–164, 1988.

[7] 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.

[8] M. Laghmouch, M. Jans, and B. Depaire, "Classifying process deviations with weak supervision," in *2020 2nd International Conference on Process Mining*. IEEE, 2020, pp. 89–96.

[9] 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.

[10] K. Yoon, Y. Liu, T. Chiu, and M. A. Vasarhelyi, "Design and evaluation of an advanced continuous data level auditing system: A three-layer structure," *International Journal of Accounting Information Systems*, vol. 42, p. 100524, 2021.

[11] M. Laghmouch, B. Depaire, N. Gigante, M. Jans, and M. Montali, "Declare moges: Model generator and specializer," in *Doctoral Consortium and Demo Track 2023 at the International Conference on Process Mining*. CEUR-WS. org, 2023.

[12] J. Mendling, H. Leopold, H. Meyerhenke, and B. Depaire, "Methodology of algorithm engineering," *arXiv preprint arXiv:2310.18979*, 2023.

[13] 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*. IEEE, 2007, pp. 287–287.

[14] R. De Masellis, C. Di Francescomarino, C. Ghidini, and F. M. Maggi, "Declarative process models: Different ways to be hierarchical," in *International Conference on Service-Oriented Computing*. Springer, 2016, pp. 104–119.

[15] I. Donadello, F. Riva, F. M. Maggi, and A. Shikhizada, "Declare4py: a python library for declarative process mining," in *Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2022 co-located with 20th International Conference on Business Process Management (BPM 2022), Münster, Germany, September 11th to 16th, 2022*. CEUR-WS. org, 2022, pp. 117–121.

[16] A. Shanthini, G. Vinodhini, R. Chandrasekaran, and P. Supraja, "A taxonomy on impact of label noise and feature noise using machine learning techniques," *Soft Computing*, vol. 23, pp. 8597–8607, 2019.

[17] 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, publisher: Elsevier.

[18] I. F. of Accountants, "International standard on auditing 315 (revised): Identifying and assessing the risks of material misstatement through understanding the entity and its environment," 2019.

[19] ——, "International standard on auditing 400: Risk assessment and internal control," 2018.

[20] M. Kabierski, H. L. Nguyen, L. Grunske, and M. Weidlich, "Sampling what matters: relevance-guided sampling of event logs," in *2021 International Conference on Process Mining*. IEEE, 2021, pp. 64–71.

[21] G. Bernard and P. Andritsos, "Selecting representative sample traces from large event logs," in *2021 International Conference on Process Mining*. IEEE, 2021, pp. 56–63.

[22] B. Knols and J. M. E. van der Werf, "Measuring the behavioral quality of log sampling," in *2019 International Conference on Process Mining*. IEEE, 2019, pp. 97–104.

[23] Y. Sun, L. AI-Khazrage, and Ö. Özümerzifon, "Generating high quality samples of process cases in internal audit," in *2021 Business Process Management Forum*. Springer, 2021, pp. 263–279.

[24] 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, 2019.

[25] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012.

[26] D. Wei, S. Cho, M. A. Vasarhelyi, and L. Te-Wierik, "Outlier detection in auditing: Integrating unsupervised learning within a multilevel framework for general ledger analysis," *Journal of Information Systems*, pp. 1–20, 2024.

[27] D. M. Schunselaar, F. M. Maggi, N. Sidorova, and W. M. van der Aalst, "Configurable declare: Designing customisable flexible models," in *20th Int. Conf. on Cooperative Information Systems*, 2012.

[28] M. L. Rosa, W. M. V. D. Aalst, M. Dumas, and F. P. Milani, "Business process variability modeling: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 1, pp. 1–45, 2017.