The International Journal of Digital Accounting Research
Vol. 25, 2025, pp. 103-133
ISSN: 2340-5058

Rule induction from exceptions and outliers for continuous auditing

Tojiboyev Nuriddin. The College of New Jersey, USA, tojiboyn@tcnj.edu

Abstract. This manuscript proposes using a feedback loop built in a form of rule induction from confirmed instances of misstatements identified through exception and outlier analyses to update existing rules or create new ones for additional filter discovery. This addition to the analytics of the continuous audit informs auditors about potential misstatement risks they might not be aware of and suggests updates and new filters that could be integrated into the transaction verification module. The proposed artifact enhances the use of risk scores in transaction verification by better reflecting the underlying causes of misstatements. Analyzing confirmed outlier instances with rule induction algorithms enables auditors to develop additional helpful rules to identify more exceptions—rules they may not be aware of. I believe that these extensions could become an important part of the continuous audit framework.

Keywords: Rule induction, exception selection, outlier analyses, continuous audit, audit analytics.

1. INTRODUCTION

The continuous auditing framework (Vasarhelyi & Halper, 1991) initially consisted of rule-based analytics derived from business processes and internal controls. Under this early version of the framework, rule-based analytics use specific metrics to generate alarms about possible misstatements in business processes for the attention of responsible personnel. Later designs of data-level implementation of continuous auditing (Kogan et al., 2014; Yoon et al., 2021) incorporated the analytical monitoring of anomalies and outliers into the framework. Transaction verification and analytical monitoring report exception and anomaly alarms to the responsible

DOI: 10.4192/1577-8517-v25_4

Submitted March 2025
Accepted August 2025

personnel. An auditor may interpret the exception alarms as his risk awareness because filters to identify these exceptions are developed from the auditor's knowledge about certain risks due to the effectiveness of business processes and internal controls. However, anomaly or outlier alarms do not represent possible causes of the misstatement but rather flag accounting items with significantly different data values from the usual ones. The review of these alarms by responsible personnel may reveal the true nature of underlying accounting items, and, importantly, whether they are fraudulent or erroneous. The continuous audit literature discusses how the outcome of reviews can be used to improve models by refining parameters (Kogan et al., 2014) or incorporating additional rules (Li et al., 2016). However, the interoperation of transaction verification and analytical monitoring is rarely discussed in the extant literature despite the numerous applications of the framework in various audit settings.

The feedback loop from the outcome of the anomaly and exception alarm investigations may serve as a critical input to the development of additional filters for transaction verification, also expanding the auditor's knowledge about previously unknown possible causes of misstatement. Furthermore, the maintenance of existing filters of transaction verification by evaluating the accuracy of filter thresholds on data fields remains an area of continuous audit that is yet to be researched. This paper proposes adding a rule induction component to the existing framework of the continuous audit between transaction verification and outlier/anomaly detection layers. This component analyzes the confirmed outliers and exceptions of the provided dataset and recommends additional possible risk filters for transaction verification and new thresholds for existing filters. The proposed component is evaluated through its application on a payroll dataset from a US non-profit organization.

Motivation for research

The latest continuous audit models include a transaction verification module that tests accounting items using predefined rules and an outlier/anomaly monitoring that extracts statistically extreme items (Kogan et al., 2014; Yoon et al., 2021). Auditors are not expected to have absolute knowledge about all possible causes of misstatement. The outlier/anomaly monitoring module serves as a "sweeping" layer on top of the transaction verification for the possibility of missing essential filters. According to Kogan et al. (2014), transaction verification and analytical monitoring

are separate automatic components that detect exceptions and anomalies and report them to responsible enterprise personnel (Figure 1). The authors discuss the role of the "Pseudo-Real Time Error Correction" in the framework design and use review outcomes of reported alarms as input for error correction. They observe the change in performance of the analytical models they chose to monitor selected business processes and find out that models with error-correction features perform better at detecting business process problems than models with no such features. The authors build their models as flows of aggregate data values. However, outlier detection algorithms can also be applied to transaction-level data to extract transactions with anomalous data values.

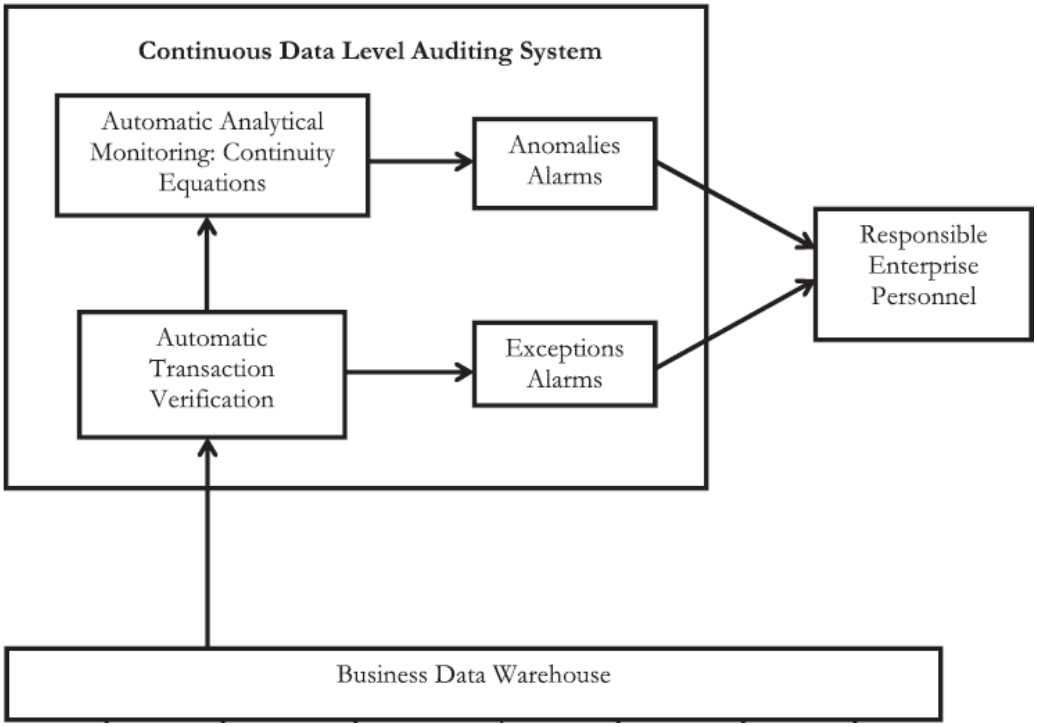

Figure 1. The architecture of the continuous data level auditing system (adapted from Kogan et al., 2014)

In another paper on the implementation of continuous audit, Li et al. (2016) discuss the expansion of transaction verification through rule addition. They use the RIPPER[1] (Cohen, 1995) learning algorithm to generate new rules based on the positive instances of misstatements confirmed during the audit investigations. This

[1] Repeated Incremental Pruning to Produce Error Reduction

approach generated 30 new rules from the instances of misstatements revealed during investigations. Although it is plausible that new rule addition to transaction verification is studied extensively by authors, they limit the feedback loop of the model to the cycle of exception identification and review. The authors do not extend the source of feedback to the analytic monitoring of anomalies. One can argue that these exceptions were already identified through some business rules and internal controls and might not provide much useful additional information about what risks are ignored by the transaction verification component of the model. The transaction verification component can be maintained and updated more accurately by incorporating the feedback from the positive instances of misstatements identified by the analytical monitoring component of the continuous auditing framework (Figure 2). The feedback about confirmed misstatements from anomalies and outliers may provide more knowledge about other business rules and controls missing in transaction verification.

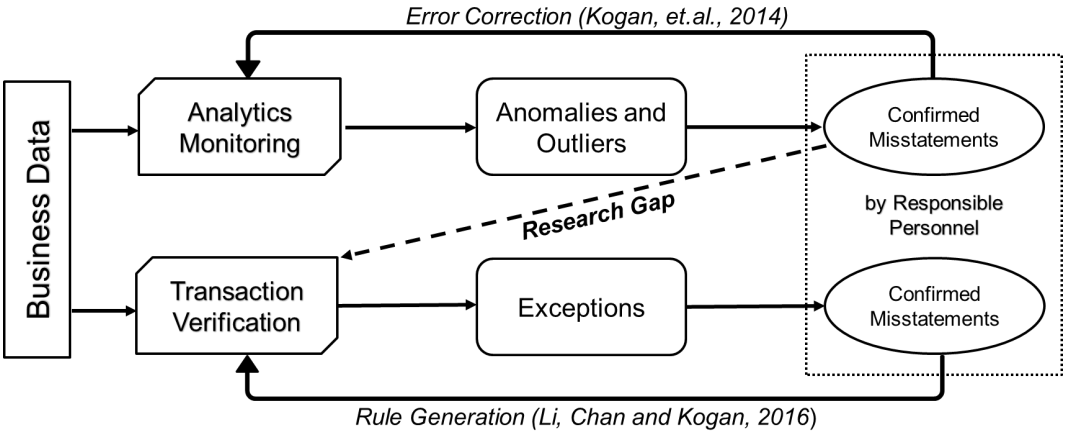

Figure 2. Feedback to transaction verification from confirmed misstatements of anomalies and outliers

The complexity of the continuous audit framework, which includes several identified modules, presents a challenge in monitoring and managing interactions among these modules. Observing these interactions requires researchers to evaluate models over multiple periods, gather feedback from each period, and incorporate that feedback into subsequent periods. For example, Kogan et al. (2014) are the first to identify separate transaction verification and analytical monitoring modules within the continuous framework and to test models using multiperiod data. Yoon et al. (2021) expand this framework by defining separate modules for control

violations and for detecting instances that fall outside predefined rules. However, the complexity of models in each study prevents the implementation of a feedback loop from other modules to transaction verification throughout the periods. Li et al. (2016) focus on improving exception prioritization through rule induction but do not incorporate an analytical monitoring module into their studies, thereby missing an important part of the feedback loop. Therefore, the interaction between the outcome of analytical monitoring and transaction verification through the discovery of new effective rules remains unexplored in the continuous audit literature. The current paper addresses this gap by including labeled instances of the investigated outliers in the training of the rule induction models for a multi-period setting. Positive instances of the outliers that are not covered by existing rules force training algorithms to induce new rules if user-imposed risk criteria are met in the dataset. The new rules become a part of the transaction verification module in subsequent periods.

Learning from confirmed misstatements of outliers and anomalies is essential for the multi-period implementation of the continuous audit framework. This information would allow the auditor to add new rules to the transaction verification and catch more suspicious accounting items in the transaction verification layer. More importantly, this learning would also enhance the auditor's understanding of misstatement risk by identifying novel, high-risk subspaces within the dataset fields that were previously unknown to the auditor. Additionally, expanding knowledge on possible causes of misstatement may allow the implementation of additional preventative internal controls in the audit setting.

Risk Awareness Exploitation vs. Risk Exploration

Although not explicitly stated in most continuous audit framework applications literature, the allocation of audit resources to the exploration and exploitation of misstatement risk is documented in prior continuous audit research (Kogan et al., 2014; Yoon et al., 2021). The transaction verification layer of the continuous audit framework can be viewed as an exploitation module since it aims to minimize the loss from misstatements by implementing the auditor's understanding of risk through predefined filters to achieve immediate goals. On the other hand, the outlier/anomaly monitoring layer serves as an exploration module to gather additional knowledge about misstatement risk. Chychyla (2014) argues that while statistical models should use limited audit resources effectively, they also should be

able to learn from previous period data and update themselves to achieve better performance. Specifically, these models are expected to learn more about the underlying distribution of data attributes to find new possible causes of misstatements. The competition for the limited audit resources between transaction verification and outlier/anomaly monitoring can be explained as an exploration-exploitation trade-off. At the auditor's discretion, it is the portion of available audit resources to dedicate to each continuous audit framework module (transaction verification and outlier/anomaly detection). The successful implementation of the framework requires a healthy balance in this trade-off. The overuse of audit resources in transaction verification may result in wasting resources investigating less risky accounting items caught by filters as exceptions, while missing out on learning critical rules from outlier detection. Contrarily, an auditor who overuses resources on outlier monitoring may receive some useless rules from the outlier detection model while not investigating the risky items that existing risk filters have caught. Thus, the auditor must plan appropriately to identify risky items using risk filters while also expanding his risk awareness in the same audit setting.

Rule Induction

The auditor's knowledge about the outcomes of investigations conducted in previous periods plays a crucial role in identifying similar misstatements in future periods. Investigating exceptions and outliers adds both positive and negative class labels to the previously unlabeled dataset, thereby reflecting the auditor's insights from these investigations. The positive instances in a labeled dataset can be generalized into machine learning models using training algorithms. Some audit investigations focus on outliers that are not covered by the current rules of automatic transaction verification. Identified positive instances of outliers can be included in the labeled dataset to help train superior models to detect similar exceptions in future periods. These models are superior to the previously used rule set of the transaction verification in that they represent possible positive instances of misstatement that were not represented in prior transaction verification. However, some machine learning models, such as artificial neural networks or support vector machines, work like "black boxes" and do not allow auditors to develop a rational understanding of the newly identified risks. Alternatively, CN2 (rule-based classification) and C4.5 (decision tree) rule induction algorithms allow auditors to understand identified positive instances of the dataset as rules. This

understanding is crucial for identifying missing internal controls and proposing new controls in a business setting represented by the dataset.

2. EXCEPTIONS VS. ANOMALIES

In continuous auditing, exceptions and anomalies are critical for identifying risks and ensuring compliance in a real-time basis. While both signal deviations from expected business operations, they differ in their origins, detection methods, and implications. Thus, separate modules are developed and maintained for each of these high-risk categories in a continuous auditing framework. However, the use and understanding of concepts of exceptions and anomalies are broader than the continuous auditing framework and are relevant in the general audit analytics domain.

Exceptions

In the literature of audit analytics (Issa, 2013; Issa & Kogan, 2014; Li et al., 2016; Yoon et al., 2021), transactions or activities that explicitly violate predefined business rules, internal controls, or regulatory requirements are known as exceptions. The identification of exceptions requires rule-based risk filters, which are performed by the automated transaction verification module of the continuous auditing framework. The rules for the risk filters could be derived from essential business rules or internal controls whose violation suggests an increased risk of misstatement. However, extant literature (Alles et al., 2006; Alles et al., 2008; Brown-Liburd et al., 2015) documented that the implementation of rule-based risk may result in an overflow of exceptions. While some scholars (Issa, 2013; Issa & Kogan, 2014; Li et al., 2016) propose prioritization of exceptions for investigations through aggregate risk scores regardless of their cross-similarities, Tojiboyev (2022) proposed selection algorithms to keep the investigated exceptions as diverse as possible.

Anomalies

Some other transactions, on the other hand, represent unusual patterns or behaviors that deviate from historical trends or expected norms but do not necessarily break predefined rules or internal controls. Anomalies often indicate emerging risks of misstatement or operational inefficiencies that may not be captured by traditional rule-based controls. These anomalies are detected using advanced data analytics, machine learning, or statistical techniques. One such approach is used by

Thiprungsri and Vasarhelyi (2011) to detect fraudulent life insurance claims by implementing cluster analyses on anomalies. In their proposed model, items with similar characteristics are grouped together and members of the clusters with smaller sizes are selected for further investigations. A similar application of clustering analyses was conducted by Kim and Vasarhelyi (2024) using DBSCAN to detect potentially fraudulent wire transfers of an insurance company. Nonnenmacher and Gómez (2021) provide a systematic review of prior research that utilizes unsupervised anomaly detection in auditing, including above mentioned clustering algorithms.

A more recent design of the advanced rule-based continuous data level auditing system proposed by Yoon et al. (2021) introduces an additional module to identify unusual transactions that have been previously recognized in predefined rules. This additional module differentiates between routine and non-routine transactions, organizing them for further analysis in groups. The authors suggest that the rules of their proposed module align with an organization's typical business activities and transactions, meaning any deviations from these rules indicate unusual transactions. One such approach that maps the possible variations of routine business processes is known as process mining in the audit analytics literature. It can be used to identify transactions that deviate from the usual process flows and to evaluate the related internal controls with these deviations (Duan et al., 2024; Wang et al., 2022).

3. CAUSES OF UNCAUGHT EXCEPTIONS

The automatic transaction verification of the continuous auditing framework is the set of audit filters that are developed from business rules, internal controls, and the auditor's understanding of the risk of misstatement. This module should generate alarms if a record of a transaction or an accounting balance does not comply with a rule or control or is deemed to represent a possible cause of misstatement. Each audit filter in the transaction verification should assess the population of accounting records regarding a separate possible cause of misstatement. However, there is some likelihood that the transaction verifications will fail to detect all material misstatements since the filters of the transaction verification module are built on the auditor's limited understanding of risk. Although auditors are expected to be aware of the risks of material misstatements, some risks might not be material individually but can be material cumulatively when aggregated. Moreover, it is not practical to create filters to cover all risks of material misstatement, as some

procedures are more manual than analytical (e.g., inventory count, assessing tone at the top, account balance confirmations, etc.). The inability to detect a certain misstatement as an exception is possible for several reasons, as shown in Figure 3.

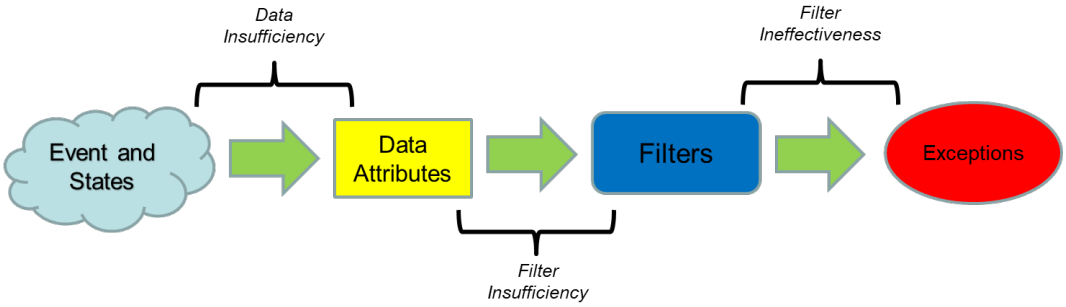

Figure 3. Causes of undetected exceptions by transaction verification

Data Insufficiency

Some misstated accounting items might not be discriminable with available data attributes. In such cases, the lack of essential data features does not allow to construct a filter that identifies these misstated accounting items as exceptions. The same issue does not permit any learning algorithms to accurately classify them as separate classes or mark them as outliers or part of the anomalies on aggregate levels. Although these cases are sporadic in practice, they may pose a significant risk of misstatement to the accounting records of transactions or balances. Accounting Information Systems must be built to collect relevant data about economic events and accounting balances to cover the risk of misstatement regarding assertions in financial reports. Additionally, users of the continuous audit framework must be careful during the data preprocessing to avoid dropping the essential data fields from the dataset that could discriminate risky accounting items.

Filter Insufficiency

The set of filters aims to detect exceptions that violate certain business rules or internal controls or have specific risk characteristics. Auditors may also expand the filter set in transaction verification with additional filters that help to decrease the overall audit risk. The additional filters are not necessarily derived from business rules or internal controls but decided by the auditor depending on his understanding of the audit setting. The collective set of filters in automatic transaction verification represents the auditor's awareness of the risk of material misstatement.

There might be certain audit risks that the auditor is not aware of in any audit setting. These risks can usually be managed through reviewing accounting items whose data attributes present anomalous values, identifying accounting items as outliers. If these outliers were not selected for review as exceptions, the analyses and investigation of outliers (that are not exceptions) help to discover risks that were not covered by filters and might be the basis to recommend adding certain new filters to the automatic transaction verification.

Filter Ineffectiveness

The filters in the automatic transaction verification are derived from business rules, internal controls, and the auditor's understanding of the risk of material misstatement. Each filter aims to detect misstatements that possess certain audit risks. A script of a filter logic that represents the rule or control is applied to values of required data attributes, and a respective filter risk score is generated for each accounting item. The risk score accuracy is dependent on the thresholds and weights used by a filter in calculating the risk score. The filter's inability to detect an accounting item that poses the same audit risk as a material and significant one, and is targeted by the filter, may be due to inappropriate threshold values used in the filter function. Setting the threshold levels too high makes filters ignore records of riskier-than-usual accounting items.

Another possible reason for filter ineffectiveness is the inappropriate allocation of filter weights. Assigning less weight to a filter significantly reduces its contribution of the filter to the suspicion score generated by automatic transaction verification. Issa (2013) proposed the allocation of filter weights through survey results by conducting it among accounting practitioners. As conducting such surveys on a regular basis is not feasible in the continuous audit framework, there is a need for a methodology that automatically reallocates filter weights when necessary. As audit settings change, the reallocation of filter weights might be crucial for a continuous auditing framework. This paper explores the possible resolution of filter insufficiency by developing additional rules using rule induction algorithms applied to confirmed outliers. This solution also recommends appropriate thresholds for some of the existing transaction verification rules, partially addressing the ineffectiveness of the filter.

4. RULE INDUCTION FROM CONFIRMED OUTLIERS

Generating rules for filter recommendations from confirmed outliers' data requires positive instances of outliers to be labeled as "Misstated". The investigated outliers' negative instances and all inliers must be marked as "Regular" (Figure 4). The confirmed positive instances of the exceptions can also be labeled as "Misstated" and added to the positive instances of the outliers to check the accuracy of thresholds in filters that caught these exceptions. The negative instances of exceptions are added to the "Regular" category and represent "safe" data sub-spaces that do not require investigation. For each input data provided, the rule induction algorithm produces a set of rules that can be recommended as filters for the transaction verification or the rules whose thresholds can be used to evaluate the accuracy of the filter thresholds.

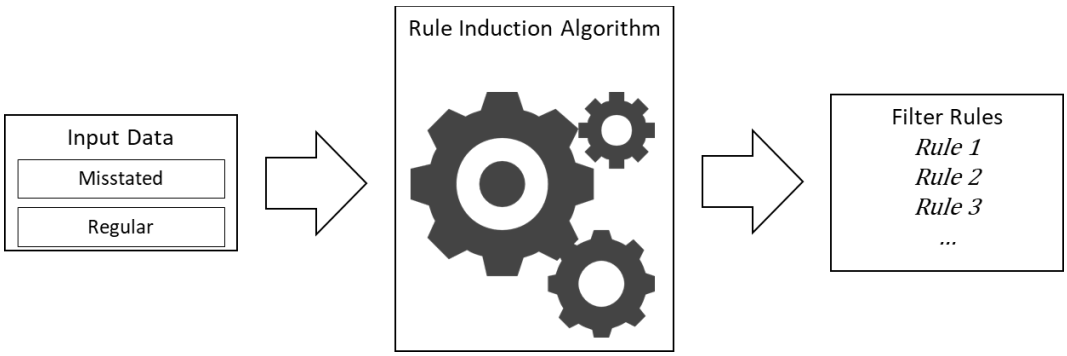

Figure 4. Input data for rule induction from confirmed outliers

While rule induction for the following period (*t+1*) from a single period of data (t) is possible, the use of multiple past periods of data *(t-n, …, t-3, t-2, t-1)* with labelled confirmed outliers increases the accuracy of the rules due to more extensive data input. The moving window approach on past periods' data can be utilized in such circumstances, increasing the rule accuracy by allowing the model to feed data from multiple periods to the rule induction algorithm and keeping the produced rules "up-to-date" by limiting the data to the recent periods. The unavailability of past period labeled data may force the user to implement the expanding window approach during the initial stages of implementation, until the intended window size is reached. Figure 5 demonstrates the moving window approach-based model with a three-period window and an expanding window in the first two periods. The ruleset for each period represents a separate rule induction process and may require the adjustment of model parameters.

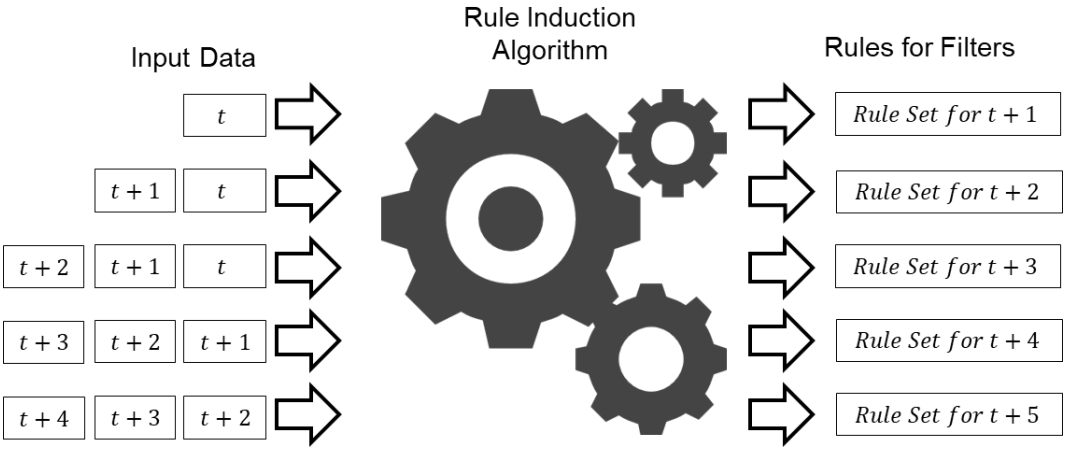

Figure 5. Moving window approach for data input to rule induction algorithms

The data analytics literature offers several rule induction algorithms (LEM1, LEM2, AQ, CART, C4.5, RIPPER, CN2, etc.) that can be used in this model. Each algorithm differs from another in its steps of reaching the intended ruleset and parameters that the user must provide to execute. Aside from the technical parameters of any chosen rule induction algorithm, following additional parameters require auditor judgment and decision to implement the model successfully.

Window Size (n). The recurrence of accounting transactions produces similar or sometimes identical accounting records. If an error or a fraud enters the accounting systems as a recurring transaction, it may persist in the records until it is detected and corrected. Using data from several past periods allows learning algorithms to generate rules that can accurately identify such misstatements. However, if the value of *n* is too large, the data from older periods would be included in the input data, pushing algorithms to focus on detecting issues from older periods but not in recent periods. Having too small *n* may not provide sufficient positive instances in the data to learn rules for discriminating underlying misstatements, resulting in an empty ruleset for the intended risky subspace.

Maximum Rule Complexity (max_depth). The complexity level in the ruleset defines the practicality of the knowledge expected to be derived from these rules. Therefore, the complexity of each rule in the ruleset should be limited to a few data fields and thresholds. Having an overcomplex ruleset may result in model overfitting and increase the influence of noise in modeling. On the other hand, if

the complexity requirement is very low, algorithms may fail to generate any valuable rules due to underfitting.

Maximum Negative Class Ratio (max_class_ratio). The exceptions reported by transaction verification are believed to contain some misstatements. However, it is unrealistic to expect all exceptions reported by a certain filter to be misstated. The responsible personnel investigating the reported exceptions will need to devote some effort to ensure that some of the exceptions are not misstated. The expected ratio of negative exceptions to positive misstatements plays a significant role in rule induction. This ratio can be interpreted as the number of exceptions the auditor is willing to investigate to find a single misstatement. The lower value of this ratio would ensure that rules are generated for a group of misstatements that share common risk features. Having the ratio value too low would make the ruleset ignore certain risky subspaces that do not have the error rate required by the user (underfitting). The higher this ratio is, the more individual misstatements may affect the ruleset. A high ratio value would make rules very specialized (overfitting), generating very complex rules. As an alternative measure, an expected misstatement rate can also be utilized, representing the portion of the exceptions detected by a filter that the auditor expects to be misstated.

Minimum Positive Instances (min_posititives). The auditor implementing the proposed model would also decide the minimum number of errors to be caught by each rule in the ruleset. This parameter would prevent generating a separate rule for each extreme case of misstated outliers. The threshold value of the parameter needs to be consistent with the parameter value of the number of periods in the moving window. An inconsistent parameter value compared to the moving window size may result in model overfitting or underfitting, as was explained in previous paragraphs.

Figure 6 illustrates the rule induction example through a decision tree algorithm from a dataset used for the model evaluation later in this chapter. Each node of the tree represents a possible rule: data field, threshold, number of instances (samples) on the node, the distribution by classes and the predicted class for each instance on the node. This example demonstrates how the Maximum Rule Complexity, Maximum Negative Class Ratio, and Minimum Positive Instances interplay in rule induction. While Maximum Rule Complexity of *max_depth=3* allows having rules with no more than three attributes, Maximum Negative Class Ratio of

max_class_ratio=1 allows pruning the right branch (*Incentive≥14093.36*) of the decision tree up to the second node. This parameter would allow combining the lower two nodes of the right branch of the decision tree (*Incentive≤22230.25* and *Incentive≥26159.86*) to a single rule despite the other negative instances (*Incentive≤26159.86*) on this node.

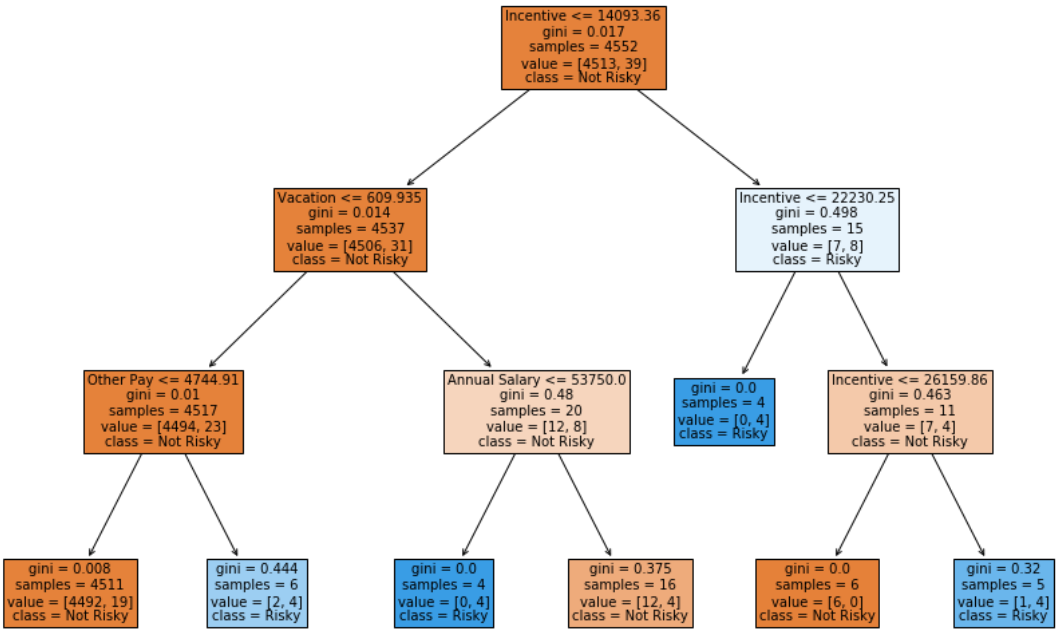

Figure 6. Interactions of maximum rule complexity, maximum negative class ratio, and minimum positive instances parameters

5. EVOLUTION OF RULES IN MULTIPERIOD IMPLEMENTATION

Depending on the values for model parameters, the rule induction algorithm may generate a different set of rules throughout the periods. A specific rule in a ruleset may evolve differently during the multiperiod model implementation, representing the periodic changes in data distributions by data attributes used by the rule. The following are examples of the changes that a rule in a ruleset may see.

Changes in Threshold. The simplest possible recommended change from a rule induction algorithm for a rule in a ruleset is a threshold value change for a particular data attribute. This recommendation is given as a result of appearing or disappearing datapoints closer to previously set threshold values from the incoming or outgoing period data in a moving window and suggests the boundary of the filter be moved along the data attribute of the threshold.

Discovery of New Rule due to Changes in Distribution. As business and accounting systems evolve, the accounting trends also change over time. The accounting systems may start collecting data about emerging new audit risks. As less noise and sufficient data about the new possible causes of misstatements are fed to the rule induction, algorithms begin to capture the general insights of high-risk subspaces where newly emerged risks reside. The accumulation of enough data points in a specific high-risk subspace allows algorithms to report this subspace as a new rule to the ruleset.

Expiration of Existing Rules due to Changes in Distribution. As an opposite of the previous, some possible causes of misstatement may discontinue to appear in accounting records and business processes due to the launch of preventative internal controls. In such cases, certain exceptions may cease to be reported by filters, making these filters obsolete. Naturally, rule induction algorithms would also remove the corresponding rule of the filter from the ruleset.

Merging of Rules due to Rule Generalizing. Due to changes in the distribution of data points by data attributes used in the existing rules of a ruleset, rule induction algorithms may combine several rules into a single rule. Figure 7 visualizes the dynamic of the rule merging in a synthetic dataset that I created to visualize the phenomenon. The synthetic dataset presents a rule-based solution to a two-class classification problem subject to the constraints introduced by user-provided parameters.

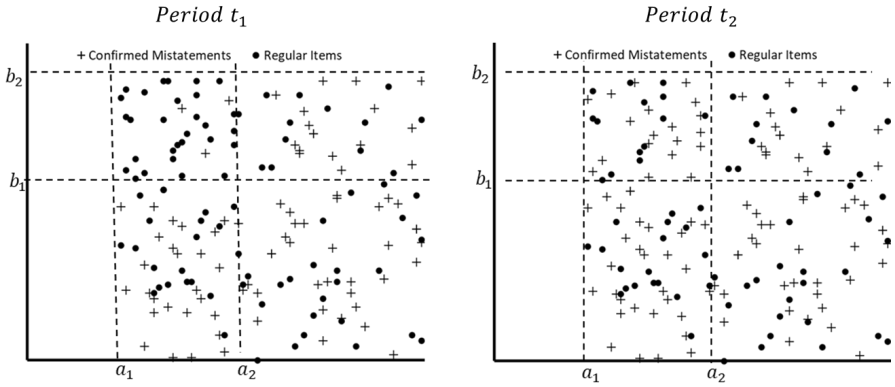

Figure 7. Merging of rules due to a rule generalization

Having too high a negative class ratio in $a_1 < x_1 < a_2 \,\&\, b_1 < x_2 < b_2$ makes this subspace not be recognized as risky by the rule induction algorithm in $Period\ t_1$. However, in $Period\ t_2$, the data from the new period on confirmed misstatements (and possibly outgoing data of an old period on negative instances) in this subspace significantly change the ratio value. Pushing the ratio value below the threshold makes the rule induction algorithm recognize the region as a high-risk subspace. However, as the other three adjacent regions are still considered high-risk, the algorithm returns the most general Rule for all four regions, producing $a_1 < x_1 \,\&\, x_2 < b_2$. The merging of rules is common during the expanding window phase of the moving window approach due to the limited number of positive instances in the initial periods.

Split of Rules due to Rule Specializing. The opposite of the previous case may also exist as a negative class ratio of a region of a particular high-risk subspace becomes lower than the ratio threshold, while other parts of the subspace keep their ratios over the threshold set by the auditor. This may also happen if the auditor decides to change the size of the moving window.

6. EVALUATIONS OF THE MODEL

Dataset and error labelling

I use a nine-month payroll dataset from a not-for-profit US organization to evaluate the model proposed (data dictionary provided in Appendix A). The dataset holds data on twice-a-month-issued paystubs of employees for each payroll period. The organization may pay its employees for several possible reasons: Regular Salary, Overtime Pay, Vacation Pay, Incentive, Tuition Reimbursement, and Other Pay. Payments to employees are delivered through either issued checks or direct deposits to their personal checking accounts. Additionally, the dataset also provides information about employees' assigned periodic and annual salaries. The sources of earnings and payment methods constitute a periodic accounting trend, making the dataset an ideal candidate for evaluations of the model.

To label some subspaces of the dataset as risky, I used the clustering of outliers as the clustering algorithm allows to divide the space into separate subspaces using the proximity of instances to each other. The K-nearest neighbor algorithm applied to the dataset provided 268 outliers. I extracted these outliers and used K-Means clustering to divide the outliers into 20 clusters with comparable subspaces. 11 of

20 clusters were labeled as "Misstated" instances, i.e., errors that the proposed model needs to learn and predict. Error labeling through clustering of outliers allows to mark certain subspaces of the dataset as high-risk and challenges the model to identify the boundaries of these high-risk subspaces rather than just predicting class values (which is widely used in the literature of machine learning applications in audit). The model is evaluated through its ability to generate reliable rules for the marked risky subspaces, discriminating the labeled errors from regular items. The auditors are more interested in the ruleset (specifying the high-risk subspaces) generated by rule induction rather than the accuracy of predictions for individual instances.

Model implementation

Once errors were labeled, I created nine fragments of the dataset, each holding the data for a month (two periods of pay) and representing a period, assuming the payroll dataset follows a monthly seasonal trend. For implementing the proposed model, I used a six-period moving window for input data selection, utilizing an expanding window for the first five periods. This window size allows the model to demonstrate its performance during both expanding and moving window phases, as rule induction algorithms may behave differently during these phases. CART[2] (Breiman et al. 1984) was used as a rule induction algorithm to generate rulesets for high-risk subspaces because of the availability of the package in Python and the script's reliability certification. A recursive rule growing procedure with parameters of *max_dept = 3, max_class_ratio = 1,* and *min_positives = 3* was applied.

A greedy heuristic rule growing (as described in Appendix B) using the decision tree algorithm CART was implemented to obtain a ruleset for each period of the dataset. Table 1 describes the results of the CART runs on the periodic data in the form of rulesets, rules, and rule thresholds. As the errors were labeled from the list of outliers rather than following the audit assertions, some filters may not present audit reasoning. However, most filters present relevant rules for payroll audit, such as unauthorized overpayments on regular pay, overtime pay, vacation pay, tuition pay, and other pay.

[2] Classification and regression trees

	Data Fields	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9
Filter 1	Tuition≥	75	75	65.5	65.5	65.5	65.5	67.97	67.97	68.97
Filter 2	Overtime≥	571.38	584.89	674.225	674.225	638.905	638.905	586.59	690.825	476.36
Filter 3	Overtime≥								476.36	
	Regular≤								1706.94	
Filter 4	Overtime≥					80.595			80.595	
	Regular≤					687.5			738.465	
	Vacation≤								150.305	
Filter 5	Other Pay≥		5934.6	4744.91	5039.83	4850.04	4850.04	4850	4850.04	4850.04
Filter 6	Incentive≥			14093.4	14093.4	13225	13225	13225	13225	13225
Filter 7	Vacation≥			609.935	609.935	507.835	491.48	491.48	491.48	491.165
	Salary≤			2239.58						
Filter 8	Vacation≥			1945.47						
	Salary≤			7284.38						
Filter 9	Overtime≥			325.32	325.32	328.32	328.34	328.34	338.28	338.28
	Salary≥			2132.19	2132.19	2132.19	2098.9	2126.3	2098.9	2479.17
Filter 10	Other Pay≥			178.775	178.775	178.775	178.775	280.15	167.775	228.8
	Regular≥			324.245	324.245	100.54	167.5	167.5	413.03	154.44
Filter 11	Regular≥									50000

Table 1. Filters, rules, and rule thresholds obtained from applying the recursive CART on a payroll dataset

Filters 5 through 9 show the effects of the expanding window as new rules start to appear in the ruleset as data of sufficient periods is fed to the rule induction algorithm. Filter 4 is an example of an unstable rule whose existence depends on data from a few specific periods and has a tendency to change the list of data

attributes to apply thresholds. This phenomenon might be due to the overfitting of the model and can be avoided by increasing the parameter value of the minimum positive instances for rules. However, one should be careful not to underfit and lose the rules of filters that are already reasonably fit (Filters 1, 2, 5, 6, 7, 9, 10).

The merge of Filter 7 and Filter 8 after Period 3 represents the effects of the expanding window during the initial periods, where more accurate data becomes available after Period 3. However, the merge of Filter 2 and Filter 3 after Period 8 serves as an example of merging rules due to rule generalizing, where incoming period data are more accurate than outgoing period data of the moving window. I did not observe any rule split due to a rule specializing in this model application.

The thresholds of most filters fluctuate (Figure 8), whereas the thresholds for rules of Filters 1, 5, 6, 7 stabilize in lower threshold values than their initial ones. This outcome recommends that the auditors adjust the threshold values of these filters accordingly. On the other hand, threshold changes of some filters (2, 9, and 10) fluctuate on both sides of 0%, suggesting current threshold values to be optimal.

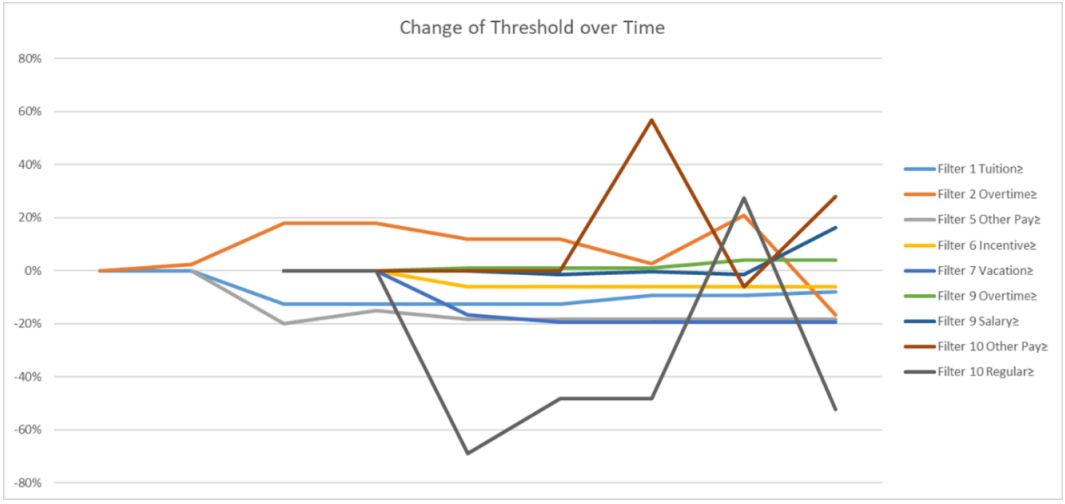

Figure 8. Changes in filter threshold over time

To evaluate model robustness, I analyze its prediction accuracy by training the model from moving window data with a six-month and applying the model to the following period data (Table 2). The results show that roughly 78% of marked exceptions in the following month are confirmed to be labeled errors. Additionally, 72% of the labeled errors were caught by the filters of the model. Although the

model accuracy fluctuates between 64.5% and 84.8%, the proportion of errors detected by the model is low during the first two periods due to its inability to generate all relevant filters because of insufficient data about errors in the initial expanding window phase. While most filters individually achieve above 60% accuracy, Filter 4 has low predictive power, which also causes its instability.

Figure 9 describes the proportions of errors caught by the model and the rate of detected exceptions that are not errors for the available periods of the dataset. These two metrics are critical for auditors in evaluating the model's performance as the former informs about the model's effectiveness, while the latter describes its efficiency. Once the model is trained with sufficient data (P4 and afterward), the filters recommended by the model detect at least around 70% of the labeled errors in the following period while wasting only up to 35% of the audit efforts on investigating the false-negative exceptions.

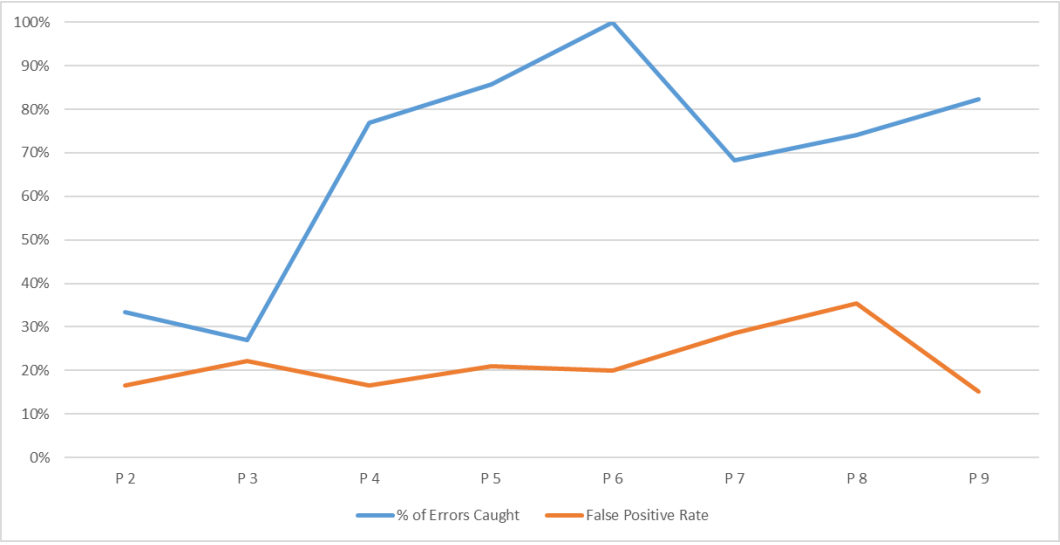

Figure 9. The portion of the errors caught and the false positive rate of exception reported by the model

Caught (Predicted)	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	Errors	Exceptions	Accuracy
Filter 1	3 (3)	4 (4)	4 (4)	9 (10)	11 (11)	5 (5)	3 (3)	5 (5)	44	45	97.80%
Filter 2	2 (3)	3 (4)	5 (5)	6 (7)	5 (7)	0 (1)	6 (11)	2 (3)	29	41	70.70%
Filter 3								1 (1)	1	1	100.00%
Filter 4					0 (2)			1 (1)	1	3	33.30%
Filter 5		0 (1)	3 (5)	2 (3)	0 (0)	2 (2)	2 (4)	0 (0)	9	15	60.00%
Filter 6			2 (2)	2 (2)	1 (1)	1 (1)	2 (3)	4 (4)	12	13	92.30%
Filter 7			0 (0)	5(9)	5 (8)	1 (4)	5 (7)	7 (10)	23	38	60.50%
Filter 8			3 (4)						3	4	75.00%
Filter 9			1 (2)	4 (5)	3 (3)	3 (3)	4 (5)	4 (5)	19	23	82.60%
Filter 10			2 (2)	5 (5)	8 (9)	3 (5)	1 (1)	6 (7)	25	29	86.20%
Exceptions	6	9	24	38	40	21	31	33	166	212	78.3%*
Errors Caught	5	7	20	30	32	15	20	28			
Errors Caught(%)	33.30%	26.90%	76.90%	85.70%	100.00%	68.20%	74.10%	82.40%	72.40%		
Accuracy	83.30%	77.80%	83.30%	78.90%	80.00%	71.40%	64.50%	84.80%	77.7%*		

*The discrepancy in accuracy is due to some errors being marked as exceptions by several filters

Table 2. Accuracy of filters on predicting error of following period

7. FACTORS AFFECTING THE PERFORMACE OF THE MODEL

The performance of the proposed model may depend on several additional factors other than algorithm parameters. Auditors should carefully review these factors for the successful implementation of the model. Some of the important factors are discussed below.

Data Preparation. Data preparation is an integral part of any data analytics-based model. Most rule induction algorithms require fields with null values to be filled or removed from the dataset. Auditors should be cautious not to introduce any additional anomalous values during the data preprocessing that significantly alter the outcome of the algorithms. Additionally, data should be prepared taking into account the peculiarities of the business cycle that is targeted by the model. This paper demonstrates the application of rule induction in a payroll cycle; however, certain considerations may change when it is applied to other business cycles.

Multicollinearity. Multicollinearity among the input data fields results in unstable rules, in which some conditions of the same rule might be defined through different data fields. In this case, when the rule generation algorithm runs, different conditions are generated for the same rule. The polymorphism of a rule in a certain ruleset causes an unstable filter that uses various data fields in different periods to point to the same risky subspace. Therefore, only one of the multiple multicollinear data fields must be kept during the data preprocessing.

Class Imbalance Problem. Some rule induction algorithms might not be able to operate effectively on datasets with class imbalance. Most, if not all, audit datasets have imbalanced classes of misstated and regular accounting records. Auditors using the proposed model must ensure that the rule induction algorithm chosen for the model can produce accurate results from datasets with class imbalances, which requires separate testing of the rule induction algorithms.

Information Gain. In supervised learning, information gained from different instances is usually considered the same. However, in an audit setting, the costs of misstatements may differ significantly by instance. Some misstatements, if not detected and corrected, may carry higher costs and more severe consequences when compared with other misstatements. Moreover, as in any audit setting, the costs of false negatives for misstatements are significantly higher than the costs of false positives. The former results in undetected misstatements, while the latter wastes

audit resources. Thus, the peculiarities of audit settings and the outcome may need to be introduced to the model to properly balance the model accuracy and cost of uncaught errors.

Feature Engineering. The performance of the proposed model is dependent on the ability of the dataset fields to discriminate high-risk subspaces. The engineering of additional data fields might be necessary if a high-risk subspace cannot be directly discriminated through the available data fields. The simplest example is weighting the field values by corresponding values of another data field (weighting the overtime amounts by the regular salary amounts). In more complex cases, more sophisticated or aggregation functions may need to be utilized to engineer new data features (e.g., extraction of weekday from DateTime formatted field, counts of duplicate payments).

Limitations and Future Research

Despite the contributions of this study, several limitations of the proposed model should be acknowledged. These limitations can serve as venues for future research to address the model's deficiencies. For example, the performance of the proposed model heavily depends on the quality of the investigations. Poor investigations of exceptions and outliers may introduce inaccurate positive and negative labels to the training dataset of rule induction algorithms, where misstated records are incorrectly labeled as non-misstated and vice versa. Inaccurate labels in training data cause the proposed feedback loop to generate ineffective rules that identify false positive exceptions while marking misstatements as false negative non-exceptions in later periods. Future research could explore the investigation process under a continuous audit framework and propose methods to ensure an appropriate investigation outcome that limits the introduction of noise to the training data of the rule induction.

Besides having accurate class labels, the availability of data features that can effectively distinguish between misstatements and non-misstatements is equally crucial for the successful implementation of rule induction methods in data-level continuous audit models. In practice, however, such discriminative features may not always be readily available in advance. In some business settings, an auditor may realize the importance of these informative features only after he gains a deeper understanding of the underlying causes of the detected misstatements. This timing issue diminishes the practical value of rule induction within a continuous audit

framework, as it introduces additional costs and delays without necessarily improving the predictive performance of the model. Consequently, the benefits of automation and real-time decision-making—key goals of continuous auditing—are undermined. Future research could address this limitation by exploring methods to engineer useful features proactively, possibly through unsupervised or semi-supervised learning techniques, to enhance the ability of audit models to properly discriminate between misstatements and non-misstatements.

And finally, the proposed model does not distinguish between the effects of learning rules and false positives and false negatives. However, in audit practices, inaccurate conclusions regarding misstatements result in different consequences that carry different costs. For instance, it is generally agreed that the costs of a false negative (marking misstated as non-misstated) are far more severe than the costs of a false positive (marking non-misstated as misstated). While false positives increase audit inefficiency by adding to the cost of the audit, false negatives undermine audit effectiveness by allowing misstatements to go undetected and uncorrected. Future research on a data-level continuous audit could explore and report on how these inaccuracies distort the induced rules and result in ineffective internal controls.

8. CONCLUSION

Transaction verification and analytical monitoring are core modules of the continuous data-level auditing system (Kogan et al., 2014; Yoon et al., 2021). While these modules select data-level records for investigations using risk filters and outlier analyses, respectively, the current continuous audit framework lacks a feedback loop between these modules. Incorporating analyses of confirmed instances of outliers and anomalies from the business rules perspective offers several advantages for continuous auditing. In a multiperiod implementation of a continuous data-level auditing system designed by Kogan et al. (2014), reported outliers are regularly reviewed by responsible personnel. The investigations of outlier instances by responsible personnel reveal the true nature of reported accounting items. The confirmations of responsible personnel can serve as useful feedback on the performance of existing filters of transaction verification. The confirmed instances of outliers that were not caught by rule-based filters can help revalidate existing rules and generate additional rules.

In this manuscript, I propose an extension to the continuous data-level auditing system by incorporating a feedback loop between transaction verification and

analytical monitoring modules. This feedback loop is created by including the labeled investigation outcomes of exceptions and outliers in the rule induction process for the transaction verification module during the following period. In the proposed approach, the positive labels of investigated outliers enable rule induction algorithms to identify rules that address the underlying causes of misstatements that were previously not covered by the rules of the transaction verification filters. The main advantage of analyzing confirmed outliers is deriving information about missing rules in the transaction verification of the continuous data-level auditing system framework. As auditors are not always expected to be aware of all risks, rule induction from outliers may provide useful insights about additional rules that are critical to the continuous audit framework.

Using this approach, I developed a data-driven model that can help identify relevant filters for confirmed instances of reported exceptions and confirmed outliers. This model incorporates several parameters (window size, rule complexity, class ratio, and instance count) that were not previously used or discussed in the prior literature. I discussed the implications of inputting inappropriate values into these parameters. "Debranching" of the decision tree was used to extract relevant rules from labeled data. The performance of the artifact was evaluated through a simulation run on multi-period payroll data. In this simulation, I marked some sets of outliers as misstatements and tested if rule-based algorithms can generate logical rules to catch these misstatements as exceptions in the following periods. If the generated rule resembles one of the existing rules (which failed to detect the outliers), the update of the thresholds of the rule was recommended. Otherwise, the new rule was added to the model starting the following period. The performance of the proposed model was evaluated through the proportion of labeled errors caught, testing the model's effectiveness, and the false-negative rate of reported exceptions by rules, testing the model's efficiency.

The model can be implemented to learn from misstated data points beyond outliers. Misstatements detected by any source can be used to train the model as long as the collection of misstatements represents a possible cause of misstatement significant to an auditor, i.e. a high-risk subspace that meets the parameters set by the auditor. Specifically, as Yoon et al. (2021) described, misstatements with unusual values can also be used to obtain a ruleset, even if these misstatements are not outliers. The successful implementation of this artifact contributes to the literature on continuous

audit by providing a means of generating and improving rules from identified misstatements.

9. REFERENCES

Alles, M., Brennan, G., Kogan, A., & Vasarhelyi, M. (2006). Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at Siemens. *International Journal of Accounting Information Systems*, 7(2), 137–161. https://doi.org/10.1016/j.accinf.2005.10.004

Alles, M., Kogan, A., & Vasarhelyi, M. (2008). Putting continuous auditing theory into practice: lessons from two pilot implementations. *Journal of Information Systems*, 22(2), 195–214. https://doi.org/10.2308/jis.2008.22.2.195

Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C. J. (1984). *Classification and regression trees.* Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8. https://doi.org/10.1201/9781315139470

Brown-Liburd, H., Issa, H., & Lombardi, D. (2015). Behavioral implications of big data's impact on audit judgment and decision making and future research directions. *Accounting Horizons 29*. 29(2), 451–468. https://doi.org/10.2308/acch-51023

Chychyla, R. (2014). Essays on accounting data differences and audit learning. Rutgers University. https://doi.org/doi:10.7282/T3H70D21 Accessed 20 February 2025.

Cohen. W.W. (1995). Fast Effective Rule Induction. *Machine Learning Proceedings.* Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, July 9–12, 1995, 115-123. https://doi.org/10.1016/B978-1-55860-377-6.50023-2

Issa, H. (2013). Exceptional exceptions. Thesis (Ph. D.)--Rutgers University. https://rucore.libraries.rutgers.edu/rutgers-lib/41484/ Accessed 25 February 2025.

Issa, H. & Kogan, A. (2014) A Predictive Ordered Logistic Regression Model as a Tool for Quality Review of Control Risk Assessments. *Journal of Information Systems*, 28 (2). 209–229. https://doi.org/10.2308/isys-50808

Duan, H. K., Vasarhelyi, M. A., & Codesso, M. (2024). Integrating Process Mining and Machine Learning for Advanced Internal Control Evaluation in Auditing.

Journal of Information Systems, 1-21. https://doi.org/10.2308/ISYS-2022-028

Kogan, A., Alles, M., Vasarhelyi, M., & Wu, J. (2014). Design and Evaluation of a Continuous Data Level Auditing System. *AUDITING: A Journal of Practice & Theory*, 33(4), 221–245. https://doi.org/10.2308/ajpt-50844

Li, P., Chan, D., & Kogan, A. (2016). Exception prioritization in the continuous auditing environment: a framework and experimental evaluation. (Report). *Journal of Information Systems*, 30(2), 135–157. https://doi.org/10.2308/isys-51220

Nonnenmacher, J. & Gómez, J. M. (2021). Unsupervised anomaly detection for internal auditing: Literature review and research agenda. *International Journal of Digital Accounting Research,* 21, 1–22. https://doi.org/10.4192/1577-8517-v21_1

Thiprungsri, S. & Vasarhelyi, M.A. (2011). Cluster Analysis for Anomaly Detection in Accounting Data: An Audit Approach. International Journal of Digital Accounting Research, 11. https://doi.org/10.4192/1577-8517-v11_4

Tojiboyev, N. (2022). *Continuous Audit Analytics Methods: The Skipper, the Stretcher, and the Looper* (Doctoral dissertation, Rutgers The State University of New Jersey, Graduate School-Newark). https://doi.org/doi:10.7282/t3-qj84-jw18

Vasarhelyi, M.A. & Halper, F.B. (1991). The continuous audit of online systems. *Auditing: A Journal of Practice & Theory* 10 (1), 110–125. https://doi.org/10.1108/978-1-78743-413-420181004

Kim, Y. & Vasarhelyi, M. (2024). Anomaly detection with the density based spatial clustering of applications with noise (DBSCAN) to detect potentially fraudulent wire transfers. *International Journal of Digital Accounting Research*, 24, 57–91. https://doi.org/10.4192/1577-8517-v24_3

Yoon, Liu, Y., Chiu, T. & Vasarhelyi, M. A. (2021). Design and evaluation of an advanced continuous data level auditing system: A three-layer structure. *International Journal of Accounting Information Systems*, 42, 100524–. https://doi.org/10.1016/j.accinf.2021.100524

Wang, Y., Chiu, T. & Vasarhelyi, M. A. (2022). Applying deep learning to detect abnormal event log traces: a non-rule-based framework. *International Journal of Digital Accounting Research*, 24, 119-140. https://doi.org/10.4192/1577-8517-v24_5

Appendix A

Data dictionary for the payroll dataset

Data Field	Data Format	Example	Input for Filter
Employee ID	Integer	475 to 21129	Filter 6
Hire Date	MM/DD/YYYY	NULL, 7/7/1975 to 9/22/2014	Filter 7
Rehire Date	MM/DD/YYYY	NULL, 12/31/2002 to 7/7/2014	Filter 7
Term Date	MM/DD/YYYY	NULL, 9/14/2012 to 9/30/2014	Filter 5
Salary	Decimal	0 to 48958.33	Filter 3, 7
Check Date	MM/DD/YYYY	1/15/2014 to 9/30/2014	Filter 5, 6, 7
Batch	Integer	1 to 4	No
Type	DIRDIP or Integer	DIRDIP or Check Number	No
Check	Decimal	0 if DIRDEP, Amount if Check	Filter 5
Dir Dep	Decimal	0 if Check, Amount if DIRDEP	Filter 5
Regular	Decimal	-10682.55 to 51041.67	Filters 1, 2, 3, 4, 7
Vacation	Decimal	0 to 34025.96	No
Overtime	Decimal	-116.57 to 17789.94	Filter 2
Incentive	Decimal	-31730.92 to 575000	No
Tuition	Decimal	-1365 to 5250	No
Other Pay	Decimal	-4016.64 to 87029	Filter 1
Annual Salary	Integer	30,000 to 1,175,000	No
Working Hours	Integer	NULL, 20 to 40	No

Table 1. Information about the fields of the payroll dataset used for the evaluation of the models

Appendix B

Greedy heuristic approach based recursive rule induction

Period 1 – Run 1. The rule induction algorithm is run on all available data for this period.

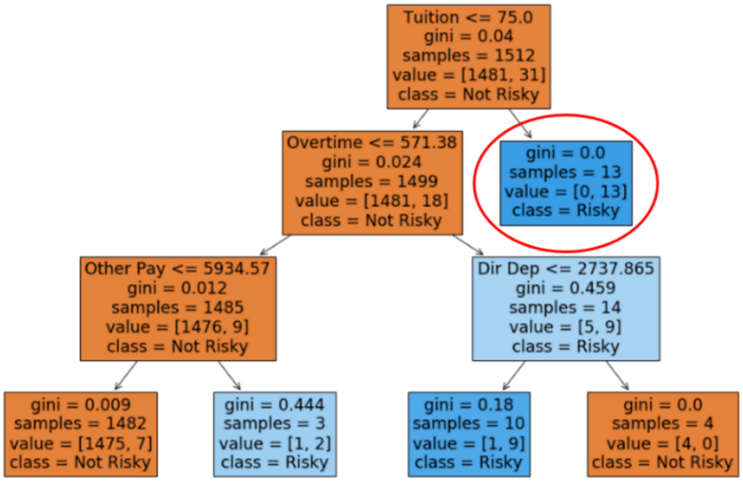

Figure 10. First Rule Induction Run for Period 1

Obtained Rule: Tuition ≥ 75.0. This rule is the simplest and has the lowest node impurity. The branch of the tree represented by this rule is extracted.

Period 1 – Run 2. The rule induction algorithm is rerun on the remaining dataset instances.

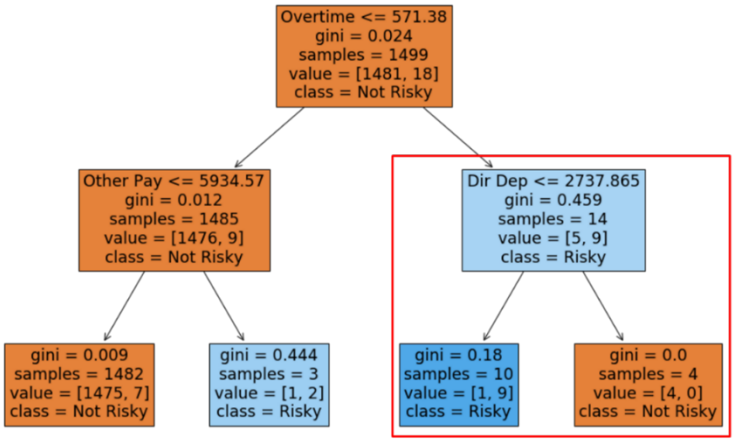

Figure 11. Second Rule Induction Run for Period 1

Obtained Rule: Overtime ≥ 571.38. This rule is simplest and matches the user parameters: maximum negative class ratio (*max_class_ratio* = 1), minimum positive instances (*min_positives* = 3) and . The branch of the tree represented by this rule is extracted.

Period 1 – Run 2. The rule induction algorithm is rerun on the remaining dataset instances.

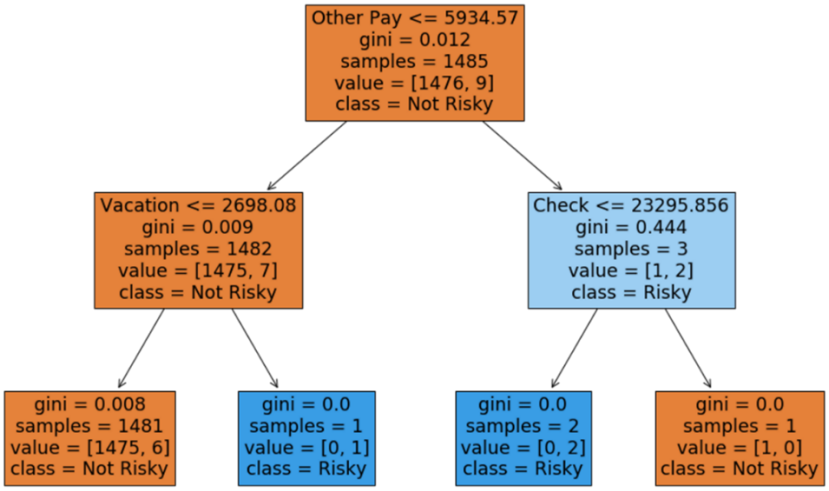

Figure 12. Third Rule Induction Run for Period 1

No rule can be obtained that matches user parameters: *max_depth* = *3*, *max_class_ratio* = 1 and *min_positives* = 3.