Outlier Detection in Auditing: Integrating Unsupervised Learning within a Multilevel Framework for General Ledger Analysis

Danyang Wei
Durham University Business School
Rutgers, The State University of New Jersey
dw561@rutgers.edu

Soohyun Cho
Rutgers, The State University of New Jersey
scho@business.rutgers.edu

Miklos Vasarhelyi
Rutgers, The State University of New Jersey
miklosv@business.rutgers.edu

Liam Te-Wierik
Grant Thornton Australia
Liam.Te-wierik@au.gt.com

Acknowledgements: We thank Won Gyun No, Helen Brown-Liburd, Hilal Atasoy, Chanyuan (Abigail) Zhang, Dong Gil Kim, and participants at University of Waterloo Centre for Information Integrity and Information Systems Assurance 2021 Symposium for helpful comments.

Danyang Wei, Durham University Business School, Department of Accounting, Durham, UK; Rutgers, The State University of New Jersey, Rutgers Business School, Department of Accounting Information Systems, Newark, New Jersey, USA; Soohyun Cho, Rutgers, The State University of New Jersey, Rutgers Business School, Department of Accounting Information Systems, Newark, New Jersey, USA; Miklos Vasarhelyi, Rutgers, The State University of New Jersey, Rutgers Business School, Department of Accounting Information Systems, Newark, New Jersey, USA; Liam Te-Wierik, Grant Thornton Australia, Audit & Assurance, Sydney, Australia.

Keywords: outlier detection; auditing; data analytics; machine learning; unsupervised learning; general ledgers

JEL Classifications: M410, M42.

Data Availability: The real dataset used in the experiment is not publicly available due to privacy policies.

Outlier Detection in Auditing: Integrating Unsupervised Learning within a Multilevel Framework for General Ledger Analysis

ABSTRACT

Auditors traditionally use sampling techniques to examine general ledger (GL) data, which suffer from sampling risks. Hence, recent research proposes full-population testing techniques, such as suspicion scoring, which rely on auditors' judgment to recognize possible risk factors and develop corresponding risk filters to identify abnormal transactions. Thus, when auditors miss potential problems, the related transactions are not likely to be identified. This paper uses unsupervised outlier detection methods, which require no prior knowledge about outliers in a dataset, to identify outliers in GL data and tests whether auditors can gain new insights from those identified outliers. A framework called the Multilevel Outlier Detection Framework (MODF) is proposed to identify outliers at the transaction level, account level, and combination-by-variable level. Experiments with one real and one synthetic GL dataset demonstrate that the MODF can help auditors to gain new insights about GL data.

I. INTRODUCTION

The General Ledger (GL) is an essential element in accounting. In the GL, financial transactions are consolidated to generate updated account balance, which are subsequently used in the preparation of financial statements. Any errors in the postings related to changes in account balances within the GL could be carried through to the final financial statement, resulting in material misstatements. Hence, it is an auditor's interest to ensure that the postings are free from material errors.

Traditionally, auditors rely on sampling techniques to select records for investigation. However, prior research finds that sampling techniques are poor at detecting low-frequency, high-risk events in large populations (Teitlebaum and Robinson 1975; Beck and Solomon 1985). Sampling can also suffer from targeting bias due to the selective nature of judgment-based sampling methodology (Blocher and Bylinski 1985; Elder and Allen 1998; Hall, Hunton, and Pierce 2000). Moreover, the investigation capacity of auditors restricts the sample size to a small proportion of the whole population, such as a few hundred out of millions of records. These arguments indicate that sampling cannot provide sufficient evidence to assure the accuracy of the records. To overcome sampling issues, recent studies propose full-population testing (Issa 2013; Li, Chan, and Kogan 2016; No, Lee, Huang, and Li 2019). Techniques, such as suspicion scoring and exception weighting, are used to test the risk of misstatement for each record based on a list of risk factors, and a suspicion score presents the assessed risk level. Higher scores indicate records with higher risk. If auditors find that the high-risk records are free of errors after investigation, they can conclude that the accounts in the GL do not contain material errors with respect to the test criteria, resulting in a greater level of confidence in the quality of the GL.

The effectiveness of full-population testing techniques significantly depends on auditors' knowledge and expertise, which are essential for creating a detailed list of risk factors and assigning appropriate weights (i.e., weighted-risk-filter techniques). Although auditors' judgment is often effective – for instance, auditors can implement a filter to detect transactions occurring on weekends, recognizing them as frequently associated with fraudulent activities – there is a risk that auditors may overlook certain risky records if they fail to identify potential underlying issues. To explore potential risks in the GL, we introduce unsupervised outlier detection methods to identify records that deviate from the normal patterns in a population. The outlier detection process is independent of human guidance, defining the normal patterns through the method itself rather than auditors' judgement. Thus, there is a chance that the method uncovers underlying abnormality in the GL that auditors do not expect.

Since no knowledge from auditors guides the outlier detection process, they must conduct an ex-post examination of these outliers to understand whether the outliers result from errors or acceptable fluctuations. Auditors rely on original variables in the records, such as dates and amounts, to assess the risk of misstatements. By contrast, outlier detection methods usually consider interaction between variables by projecting them onto a multi-variate space where the normal patterns are identified and the degree of deviation for each record is determined. When the dimensionality of the space is high, the cause of outliers would not be interpretable to humans, and examining the identified outliers directly is the only way to gain new insights.

In an attempt to encompass different types of outliers in a GL dataset, we propose a Multilevel Outlier Detection Framework (MODF) that identifies outliers at three levels: transaction level, account level, and combination-by-variable level with each level representing a specific mechanism for outlier detection. We then test the effectiveness of the framework using both real and synthetic GL datasets. For the real GL dataset, an external audit team from one of the world's largest audit firms partnered with our research team to provide the data and conduct risk assessment on the identified outliers. The risk assessment results are presented as a risk level (either low, medium, or high) for each identified outlier. Additionally, to examine whether new risk is identified by the MODF, we compare the risk levels of the identified outliers to the suspicion scores that the audit team has calculated using a weighted-risk-filter methodology. The methodology involves the development of a list of risk filters and the assignment of weights for each filter to calculate a suspicion score. Through the comparison, we find that some identified observations receive higher risk levels when the risk assessment is based on the identified outliers, as opposed to when it is based on the suspicion scores.

The synthetic GL dataset[1] is from a case study created by Ernst & Young Academic Resource Center (EYARC). The dataset contains no fraud. We test the ability of the MODF in complementing auditors' scope by simulating a specific type of seeded errors that may elude auditors' knowledge. Particularly, the variable values of the seeded errors fall into the same ranges as the records in the dataset, but the relation between the variables is changed. Since the relation is not directly observable, it is difficult to identify the abnormal relation with risk filters, which are derived from observable abnormal scenarios (e.g., transactions occurred during weekends). Nevertheless, the MODF exhibits promising performance at detecting those seeded errors. This experiment aims to demonstrate the MODF's capability in uncovering abnormalities within GL data that are typically overlooked by existing full-population techniques. The findings from both the real and synthetic datasets demonstrate that auditors can employ outlier detection methods to identify suspicious records within GL data. Through examination of the suspicious records, auditors may gain new insights regarding the risks of the GL data.

This paper contributes to audit research and practice in several ways. First, it demonstrates the feasibility, through the MODF and its implementation, of applying outlier detection methods to selecting suspicious observations in GL data. Second, we compare the MODF to existing full-population testing techniques that mainly rely on auditors' knowledge and illustrate that outlier detection methods can help auditors identify unexpected risks in GL data.

The paper is organized as follows: Section II reviews prior research in outlier detection applications and audit analytics. Section III introduces the MODF. Section IV demonstrates the

[1] Available at https://eyus.sharepoint.com/sites/EYARC/SitePages/AM-Case-Studies.aspx

framework with both real and synthetic datasets and reports the results. Section V discusses the potential uses and limitations of the MODF. Section VI concludes the paper.

II. LITERATURE REVIEW

Assessing the risk of material misstatement is a fundamental task for auditors. Analytical procedures must be performed to enhance auditors' understanding of the client and to identify areas that represent certain risks for further investigation (AS 2110.46, 2016). Auditors' knowledge is generally effective in identifying risk factors. However, research shows that auditors' reaction to fraud cues may sometimes be flawed due to the strategic nature of fraud (Wilks and Zimbelman 2004; Asare and Wright 2004; Hoffman and Zimbelman 2009). Unsupervised outlier detection, which emerged from computer science, aims to distinguish abnormal data patterns from normal ones with no prior knowledge about the abnormality. Thus, it could offer a chance for auditors to gain new knowledge about their clients from the outliers that exhibit abnormal characteristics. Despite the demonstrated effectiveness of various outlier detection methods in fields such as chemistry, medicine, and finance, limited research has examined their application in an audit context. This paper contributes to both the outlier detection and the audit analytics literature by illustrating the effectiveness of outlier detection methods in auditing.

Outlier Detection

Grubbs (1969) defines an outlier as an observation "that appears to deviate markedly from other members of the sample in which it occurs." Analyzing the outliers may reveal weakness in the generation system from which the sample was derived.

One of the most fundamental outlier detection models is k-nearest neighbors (KNN), which calculates the distance of each point to its k-nearest neighbors. Points with a significantly large distance from all other points are global outliers (Ramaswamy, Rastogi, and Shim 2000). However, such obvious outliers are not always present in real data. More commonly, outliers are not very far from a main clusters but are surrounded by fewer similar points. Such outliers are called local outliers, which are close to a main cluster but in a low-density space. Breunig, Kriegel, Ng, and Sander (2000) propose Local Outlier Factor (LOF), a density-based model to identify both global and local outliers by comparing the local density of a given point to the average density of its k-nearest neighbors, which produces an LOF value (i.e., an outlier score). A higher LOF value represents a higher likelihood that the point is an outlier.

As a simple but effective detector model, LOF has been applied to various outlier detection problems and used as a benchmark in outlier detection research. For example, Mokua, Maina, and Kiragu (2021) apply the LOF algorithm to detect anomalies in water quality data, and Alghushairy, Alsini, Soule, and Ma (2020) discuss the potential of LOF algorithms for outlier detection in big data streams.

Other well-known outlier detection approaches include probabilistic models and univariate methods. Probabilistic models first infer the distribution of the majority of data and then identify points that are not likely to belong to it. The most common probabilistic model is Gaussian Mixture Model (Zhuang, Huang, Palaniappan, and Zhao 1996). Univariate methods instead focus on a variable's extreme values and are usually applied together with visualization methods (e.g., z-scores, boxplots, and/or histograms). Probabilistic and univariate models are more effective when data is generated from a single activity.

Research demonstrates the diverse range of domains where outlier detection algorithms are utilized to identify outliers with significant implications. The primary application domain is intrusion detection. In this scenario, malicious activities in networked computer systems are identified via outlier detection (e.g., Davis and Clark 2011). Another application domain is fraud detection, where log data is used to detect suspicious records indicating fraud, such as unusual transaction amounts that might suggest credit card fraud. Srivastava, Kundu, Sural, and Majumdar (2008) use k-means clustering to create spending profiles for individual cardholders' spending amounts. An alarm is reported when a new transaction deviates from a cardholder's existing spending profile. Cynthia and George (2021) implement two unsupervised algorithms, Local Outlier Factor and Isolation Forest, and two supervised algorithms, Support Vector Machine and Logistic Regression, on a credit card dataset to observe their ability to identify fraudulent transactions. Their results show that unsupervised algorithms are "more suitable for practical applications of fraud and spam identification." Another example of fraud detection is finding fraudulent accounting in financial transactions (Debreceny and Gray 2010; Khan, Corney, Clark, and Mohay 2010; Thiprungsri and Vasarhelyi 2011; Khan, Clark, Mohay, and Suriadi 2014). For instance, Thiprungsri and Vasarhelyi (2011) apply k-means clustering to an insurance dataset and define outliers as (1) observations that are far from a main cluster, and (2) observations in small clusters. By checking those outliers, they find some suspicious insurance claims.

Overall, unsupervised outlier detection is a well-studied field that comprises a diverse range of methods to identify outliers with various abnormal characteristics. Furthermore, the application studies that show the ability of unsupervised outlier detection methods to identify meaningful outliers motivates this paper to examine what these techniques will find when used in an auditing scenario.

Advanced Data Analytics Techniques in Auditing

Audit data analytics (ADA) is defined as "the science and art of discovering and analyzing patterns, identifying anomalies, and extracting other useful information in data …for the purpose of planning or performing the audit" (AICPA 2017). The analysis of large datasets using various analytical tools is the focus of many studies where the objective is to achieve a comprehensive understanding of the data. For example, Issa and Kogan (2014) apply a logistic regression model for quality reviews of internal controls. Similarly, Zhaokai and Moffitt (2019) develop a novel Contract Analytics Framework (CAF) to help auditors conduct analyses on full populations of contracts, which are traditionally examined manually on a sample basis. Visualization is another powerful tool that can be used to analyze data and generate new insights. While visualization is commonly used as a complementary tool to illustrate data, Alawadhi (2015) describes how visualization benefits auditors throughout the audit cycle. Nevertheless, choosing the most appropriate type of information representation for a given task can be challenging, as noted by Dilla, Janvrin, and Raschke (2010). Process mining is another analytical tool to evaluate the effectiveness of internal control by scrutinizing data from event logs. These logs record computer system activities chronologically and automatically, providing valuable data that is independent of the auditee's manipulation (Jans, Alles, and Vasarhelyi 2013).

Although their data and analytical approaches vary, these studies share a common ADA objective, which is obtaining sufficient evidence to form opinions based on large datasets. The ADA can be further categorized into two groups depending on what the auditors know about the problem's characteristics. If auditors know the possible risk factors and use ADA for testing, then it is a confirmatory data analysis (CDA), whereas if auditors use ADA to "develop new hypotheses or refine current hypotheses" about their clients, then it is an exploratory data analysis (EDA) (Tukey 1977). Most ADA studies fall into the first category since guidance from auditors is provided to the ADA model to obtain the output. However, this study designs its framework with the EDA mind and demonstrates how it helps to identify new risks. Research that also focuses on EDA includes studies by Thiprungsri and Vasarhelyi (2011) and Liu (2014).

The idea of full-population examination of GL data has drawn much recent attention from researchers. Li et al. (2016) assign suspicion scores to each transaction based on violation of predefined expert rules and set a threshold for these scores to select transactions to be investigated. No et al. (2019) suggest that weighted filters be developed based on risk factors for suspicion scoring. The resulting suspicion scores serve as a proxy for auditors' judgments, and their study uses exception prioritization to further select "exceptional exceptions (Issa 2013)". In a more recent study, Freiman, Kim, and Vasarhelyi (2022) apply the MADS methodology to a real-world GL dataset and demonstrate its effectiveness. While most studies propose full-population examination methodologies within a CDA context, our study is focused on an EDA context. The new insights gained through EDA can benefit CDA by complementing its testing objectives.

III. METHODOLOGY

We propose a Multilevel Outlier Detection Framework (MODF), shown in Figure 1, to identify outliers in GL data at three different levels. For each level, a detection method is introduced to identify the target outliers. The framework incorporates four steps: data preprocessing, outlier detection, two-stage prioritization, and investigation.

Step 1: Data Preprocessing

The data used comprises all postings associated with changes in account balances within a GL, readily exportable from a firm's ERP system. The records typically require conversion into a suitable format for outlier detection. First, variables containing missing values should be eliminated during data cleaning. Furthermore, if the information contained in a particular variable can be derived from another variable, it should be omitted to prevent redundancy. Second, new variables can be created to capture specific and targeted information. The process of variable engineering should align with the objectives set by auditors. For instance, if auditors are concerned about backdating, they can create a variable by computing the discrepancy between the event date and the corresponding bookkeeping date. Third, numerical variables should be normalized to ensure their comparability. Otherwise, extreme variable values would dominate the outlier detection results. Finally, the records are reconstructed into three different datasets to detect target outliers. The first dataset consists of individual records containing all relevant variables, from which the transaction-level outliers are identified. The second dataset involves all accounts in the GL with their balances. The target outliers in this dataset are account-level outliers. The third dataset involves all individual records, but only the categorical variables are retained. The numerical variables are removed to specifically identify combination-by-variable-level (CBVL) outliers. For each observation, its categorical variable values form a combination-by-variable. The CBVL outliers aim to uncover the condition in which the variable values are frequent, but specific combinations of those variables are infrequent.

Step 2: Outlier Detection Process

After constructing the three dataset, an outlier detection model is applied to each. The model used to detect transaction-level outliers is Local Outlier Factor (LOF), in which observations that are not closely surrounded by their neighbors are outliers (Breunig et al. 2000). An outlier score is assigned to each observation, which represents its outlying degree. A necessary parameter to execute the model is the number of observations to be considered as "neighbors" for each observation. Given the fact that the optimal value is not observable, Breunig et al. (2000) recommend running the model multiple times, each with a different parameter value applied.[2] Among the multiple outlier scores for each observation, the maximum value becomes its final score.

[2] A range of 10 to 20 is often used as a rule of thumb according to Breunig et al. (2000).

The Z-score model is used to detect account-level outliers, which are accounts with extreme balances in the GL. The function to calculate the Z-score is as follows:

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where x is an account balance, μ is the average account balance of the GL, and σ is the standard deviation of the account balances in the GL. Higher Z-scores indicate higher deviation from the average.

The CBVL outliers are identified by K-Modes Clustering (Huang 1998), which is designed to cluster data with categorical variables only. The number of clusters, *k*, is a necessary parameter to execute the model. As a rule of thumb, an effective way to obtain the optimal k value is the elbow method. Initially, the model selects k frequent combinations to be the centers of k clusters (i.e., k modes). Subsequently, the distance between each observation and the center of each cluster is calculated with a dissimilarity measure[3] to determine the nearest cluster that the observation belongs to. The k modes are updated until the within-cluster difference reaches the minimum. After ranking the k clusters according to their cluster sizes, observations in the smallest clusters carry infrequent combinations.

Step 3: Two-Stage Prioritization

To select observations for further investigation, each dataset is ranked in a descending order by the output of the applied detection model, either outlier scores, Z-scores, or cluster sizes. The investigation should focus on the top observations in each dataset. Since auditors have limited investigation capacity, the number of selected observations should not be overwhelming. Another concern is that many observations selected have immaterial dollar values. To address these

[3] For each categorical variable, if an observation has the same value as the mode, the distance is zero; otherwise, the distance is one. The sum of all the categorical variable distances is the distance of an observation to a mode.

concerns, we apply a two-stage filtering process to prioritize outliers to be investigated. At stage one, a threshold for the outlying degree above which observations are notable outliers is set as a percentile for the outlier scores, Z-scores, and cluster sizes. At stage two, a dollar value serves as another threshold to filter out notable outliers with immaterial amounts. The notable outliers with amounts above this threshold are exceptional outliers at the transaction-level, account-level, and CBVL, which will be investigated in the next step. Auditors should determine the two thresholds in terms of their general understanding of the client's risk.

Step 4: Investigation

After investigating the exceptional outliers, auditors can draw a conclusion about their risks. Auditors can either use a binary label (e.g., whether the outlier is a misstatement or not) or a risk level (e.g., low, medium, or high), depending on their judgment during the investigation.

Another analysis is conducted to test whether additional insights are gained through the investigation of the outlier detection results. We argue that when auditors identify unexpected misstatements or assign an increasing risk level to an observation, they gain new risk-related insights through the outliers. In the second condition, a full-population testing technique based on auditors' judgment (e.g. the weighted-risk-filter technique) needs to be applied first so that we can compare risk assessment results based on the auditors' judgment and the MODF. When the newly assigned risk level for a given outlier is greater than the existing one, it indicates that auditors have obtained new, useful knowledge during the investigation of exceptional outliers. Moreover, if the new knowledge is applicable to other audit engagements, new risk factors may be recognized to identify similar records in the future.

IV. EXPERIMENT

This study uses two entry datasets to demonstrate the effectiveness of the framework. The first dataset is all journal entries and their associated postings to the GL for a real company for FY 2019, provided by the firm's external audit team. In total, there are 521,283 observations. The Appendix shows examples of journal entry data. After the framework identifies the exceptional outliers, the audit team investigated them and provided the risk assessment results. Additionally, the audit team also has the risk assessment results based on a full-population testing technique using weighted risk filters. Hence, we can compare the risk assessment results for the same dataset based on the auditors' judgment and the exceptional outliers.

Another dataset is synthetic and created by EYARC for a case study. It contains 37,869 journal entries of a university hotel. There are two versions of this dataset: with and without fraudulent entries. The clean version is used in this study. In order to form a better understanding of the condition where the MODF outperforms the weighted-risk-filter techniques, we contaminate the clean data with seeded errors created in a way that their variable values fall into the same ranges as the observations in the dataset, but the relation between the variables is changed. Since the relation is not directly observable, such seeded errors are difficult to identify based on auditors' judgment which focuses on observable abnormal scenarios (e.g., violation of segregation of duties).

Real-World Dataset

The 521,283 journal entries result in changes in the balances of 1,479 unique accounts. Table 1 outlines the nine variables in the data.

Step 1: Data Preprocessing

During data cleaning, we exclude the Line Description variable because all observations have missing values, and the Transaction ID variable is also dropped because that information can be derived from the other variable in the data. For variable engineering, we create a new variable called Day Difference by calculating the discrepancy between Effective Date and Created Date/Time. Such a variable is created to capture observations with abnormal lags between the two dates, indicating a delay in recording transactions. Another new variable, Keyword Count, is created based on the Journal Description variable, which is aligned with the audit team's risk assessment procedures for journal descriptions. Particularly, the team compiles a list of keywords deemed risky in descriptions, and auditors review each description containing the keywords to assess the need for further investigation (e.g., to ascertain if a description is vague). Given that the latter part of the risk assessment heavily relies on auditors' judgment, we concentrate on the initial phase of their procedure aimed at identifying journal descriptions with potential risks. Drawing on prior studies in textual analysis, which suggest that more negative words indicate a more negative sentiment (Antweiler and Frank 2004; Tetlock 2007; Loughran and McDonald 2011), we argue that a greater presence of keywords in a description signals a higher risk level. Thus, we established the Keyword Count variable to count the risky keywords in each journal description for outlier detection. An observation with a substantial keyword count is more likely to receive a high outlier score, identifying it as a significant outlier. Finally, the absolute values of Net, Day Difference, and Keyword Count columns are normalized, while the categorical variables, Account Code, Document Type, and User ID, are converted into binary variables through one-hot encoding. Table 2 provides a summary of the variable preprocessing.

For data restructure, we first separate the accounts based on their GL categories, which are generally established to divide transactions generated from different business activities, and then construct the three datasets used for outlier detection within each category. Such separation ensures that the outliers are identified in a population comprising records derived from similar business activities. Otherwise, the outliers would be less meaningful if the population is composed of records from heterogeneous business activities. A GL category can have multiple accounts, and the transactions involved in each account all belong to that category. The GL category that each account (and the transactions in it) belongs to can be obtained from the trial balance of the company. In total, there are 31 categories. The audit team selected six categories for us to apply the framework, within which most records are low-risk with respect to their risk-filter-based technique. Table 3 lists the six categories and their indices. For the three datasets constructed within each category, Table 3 also presents the numbers of transaction records (hereafter records), accounts, and unique variable combinations.

Step 2. Outlier Detection

LOF, Z-score, and K-Modes Clustering are applied to the three datasets in each primary category, yielding outlier scores, Z-scores, and *k* clusters as the respective outputs. The Python code to run LOF and K-Modes Clustering is available at scikit-learn[4] and Github[5], respectively.

Step 3. Two-Stage Prioritization

At stage one, the threshold for identifying notable outliers is set at the 75th percentile of the outlying degree, which echoes the threshold used by the audit team for the suspicion scores calculated by a weighted-risk-filter technique. The auditors set the suspicion score at the 75th percentile as the threshold for identifying high-risk records, which produced a sample of 79 records. While the threshold yields a reasonable sample size in this scenario, it may identify a large number of notable outliers that overwhelm the auditor's investigation capacity, especially when there are

[4] https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html

[5] https://github.com/nicodv/kmodes

very few observations with the same outlying degree.[6] Hence, an upper limit of 100 records is applied to the sample size. When the sample size with the 75th percentile threshold is less than 100, all the observations above it are considered notable outliers. However, if the sample size exceeds 100, only the top 100 observations are designated as notable outliers. For this study, the auditors agree that 100 is a reasonable sample size for investigation.

Regarding the stage-two threshold, the audit team suggested $10,000 as an amount below which they would consider an outlier to be immaterial for this client. Table 4 outlines the two-stage prioritization results for the three levels. The 75th percentile threshold is applied as the stage-one threshold for all account-level observations and the CBVL observations in Categories A-D), whereas the top-100 limit becomes the threshold for all transaction-level observations and the CBVL observations in Categories E-F. As Table 4 shows, there are a total of 41 exceptional records, 50 exceptional accounts, and 43 exceptional combinations to be investigated in next step.

Step 4. Investigation

The auditors first investigated the exceptional records, accounts, and combinations, and then categorized them as being at a low, medium, or high-risk level. These newly assigned risk levels are the risk assessment results based on the exceptional outliers. Table 5 summarizes the number of exceptional observations that fall into each risk level. As the auditors only spend investigation budget on medium or high risks, we considered an observation risky only when it receives a medium or high-risk level. According to Table 5, observations with medium or high risks are identified in all the three outlier detection levels. Particularly, 17.1 percent exceptional records, 82 percent exceptional accounts, and 69.8 percent exceptional combinations receive

[6] For instance, if LOF assigns a unique outlier score to each observation, the 75th percentile threshold will select exactly 25 percent of the population as notable outliers, which will be an overwhelming sample for auditors to conduct investigation.

medium or high risks. It indicates that the three detection methods in the MODF are more effective in identifying risky accounts and combinations with respect to auditors' criterion.

The risk levels of the exceptional observations are then compared to the risk assessment results based on the auditors' weighted-risk-filter technique. The audit team already developed an analytical routine prior to this study, which involves a list of risk filters and a weight to each filter. For example, one of the risk filters is transactions that occurred during weekends. For each record, auditors calculated a suspicion score using the following formula:

$$s_i = \sum_{j=1}^{n} x_{ij} w_j \tag{2}$$

where x_{ij} is a binary variable that equals one if the record i violates the risk filter j and zero otherwise, w_j is the weight of risk filter j, and n is the total number of risk filters. After the population was ranked from the largest to the smallest by the suspicion scores, the auditors set two thresholds to determine high-risk (at or above the 75th percentile), medium-risk (between the 75th and 60th percentiles), and low-risk (below the 60th percentile) records.

The existing risk assessment results based on the weighted-risk-filter technique are only available to individual records. Hence, to obtain the existing risk assessment results of the accounts and the variable combinations, we aggregate the records by their account or combination and take the maximum risk level among the records in that category to be the existing risk assessment result for that account or combination. For each exceptional observation (i.e., record, account, or combination), the newly assigned risk level is compared to the existing one. We argue that when the new level is higher than the existing one, auditors gain new risk-related insights about the GL data.

Table 6 presents the comparison results. No higher new risk level is found in the exceptional records, whereas five (10 percent) of the exceptional accounts and 29 (67 percent) of the exceptional combinations received a higher risk level using the MODF. These results indicate that auditors can gain additional insights about the GL data through our analysis of exceptional outliers, which may affect their risk assessment results. A possible explanation of the MODF identifying no additional risk in the exceptional records is that the auditors are unaware of the reason why those records are flagged by LOF. Unlike exceptional accounts and combinations that are identified as outliers due to extreme balances and infrequent combinations of categorical variable values, the exceptional records are identified in a multi-variate space in which the interaction between the numerical and the categorical variables are considered. Given the complex working mechanism of LOF, the auditors do not have an intuition about why those records are identified and can only assess the risk of the records based on their existing knowledge, resulting in identifying no additional risk.

Synthetic dataset

The EYARC dataset contains 37,869 journal entries for the fiscal year of a university hotel, which involves changes to the balances of 73 GL accounts. Table 7 lists the 15 variables in the dataset.

Seeded Error Generation

In the demonstration with the real dataset, the auditors evaluate the identified exceptional records and combinations without understanding the factors that contribute to their outlying status[7] because the mechanisms inherently employed by LOF and K-Modes clustering are not observable.

[7] Although we have a general understanding of why CBVL outliers are identified, the exact process to get the outliers is still unclear. For instance, whether a given CBVL outlier in the real dataset it is due to an infrequent combination of Account Code and User ID, User ID and Document Type, or all of these is not known.

As a result, it is still uncertain what types of misstatements can be more effectively detected by LOF and K-Modes than the weighted-risk-filter technique. To resolve this concern, we generate seeded errors that, by their nature, are difficult to identify using the weighted-risk-filter techniques to test whether LOF and K-Modes can identify such errors. Particularly, those errors have individual variable values close to the normal observations', but when the relation between those values is analyzed, the errors and the normal observations will produce different dependency structures. We estimate the probability density function of each variable in the dataset to ensure the variable values of the errors are comparable to the original observations. Meanwhile, we utilize a vine copula to describe the relation between the variables. Vine copulas are graphical models that build a dimensional dependency structure for an arbitrary number of variables, and changing the dependency structure generates dependency outliers. This generation mechanism is introduced by Steinbuss and Böhm (2021). The full multivariate probability density function can be written as follows:

$$f(x_1, \dots, x_d) = f_1(x_1) \cdot \dots \cdot f_d(x_d) \cdot c(F_1^{-1}(x_1), \dots, F_d^{-1}(x_d)) \tag{3}$$

where $f_i(\cdot)$ is the probability density function of variable i, $F_i^{-1}(\cdot)$ is the cumulative distribution of variable i, and $c(\cdot)$ is the copula model.

We do not modify estimates for the individual variable density functions. Instead, we set the dependency structure as complete independence using the *rvinecopulib* R package.[8] After synthetic observations are generated from the modified full density function, we remove observations with variable values that do not fall into the range of the corresponding variable in the dataset. This generation process allows us to produce seeded errors that have normal variable

[8] Available at https://CRAN.R-project.org/package=rvinecopulib.

values but different variable structures from the original observations in the dataset. The percentages of seeded errors are set at 1 percent, 3 percent, and 5 percent.

Step 1: Data Preprocessing

During data cleaning, we exclude seven of the 15 variables from examinations because they either (1) are derived from another variable in the dataset, (2) have a 1-to-1 relationship with another variable in the dataset, (3) have same values for all observations, or (4) require additional but unavailable information to be used. At the variable engineering step, the difference between Effective Date and Entry Date is computed and used as a new variable named Day Difference. The absolute values of Amount and Day Difference columns are normalized, while the categorical variables, Business Unit, GL Account Number, Preparer ID, and Source, are converted into binary variables by one-hot encoding. Table 8 provides a summary of the variable preprocessing.

To reconstruct the data, the accounts are separated based on the Account Class column to ensure that the population in which outliers are identified only contains records from similar business activities. In total, the dataset has 18 account classes. We select seven account classes that contain over 1,000 records for the experiment in order to have sufficient observations to estimate the full density function. The numbers of records, accounts, and variable combinations in each account class are listed in Table 9. Based on the seeded 1-percent, 3-percent, and 5-percent errors added to each account class, two datasets are constructed for the detection of transaction-level and CBVL outliers. The numbers of records and the unique combinations after the seeded errors are added are also shown in Table 9. The seven account classes involve 34,257 original records, representing over 90 percent of the dataset.

Step 2: Outlier Detection

We apply LOF and K-Modes to each of the account classes to identify transaction-level and CBVL outliers. Account-level outliers are omitted for two reasons: First, as a univariate method, Z-score employs a straightforward operational principle to identify outliers, so auditors could fully understand why an account is identified as an outlier. Second, except for SG&A, each of the account classes used contains only few accounts, a population too small for outlier detection.

Step 3: Prioritization

In this experiment, we only implement a one-stage prioritization because the materiality amounts for a stage-two prioritization are not available. Two different stage-one thresholds are utilized to select outliers for investigation: top 100 observations and top 200 observations. To simulate a more realistic application scenario, we do not use the actual number of seeded errors as the threshold since auditors typically have no access to that information. At the transaction level, the outliers selected are the top 100 (or 200) records with the highest outlier scores. At the CBV level, the outliers are the top 100 (or 200) combinations in the smallest clusters.

Step 4: Investigation

Table 10 exhibits the accuracy rates that are calculated by dividing the number of actual seeded errors in an investigation sample by the sample size. We also determine the potential maximum accuracy rate for each investigation sample by dividing the total number of seeded errors added by the sample size (either 100 or 200) and present the results below the actual accuracy rate in Table 10. By comparing the two rates, we can implement a more precise evaluation because the actual number of seeded errors can be less than the threshold. For instance, if the total number of seeded errors is 20, then an accuracy rate 0.18 under the 100 threshold is high. As Panel A shows, the inclusion of an additional 100 observations in the investigation sample does not lead to a higher accuracy rate for LOF. Furthermore, for the four account classes that have an actual accuracy rate that is close to the potential maximum rate at the 1-percent setting (Accrued Payroll, Inventory, Sales-Dining, and Sales-Other), the difference between the two rates increases at the 5-percent setting. These results indicate that LOF may not be an effective option to identify dependency outliers. In contrast, Panel B shows that the accuracy rates for all account classes are high. The largest difference between the actual rate and the potential maximum rate (1 – 0.65 = 0.35) occurs in the account class Sales-Hotel with the 3-percent setting and the top-100 threshold. However, as with LOF, increasing sample size in K-Modes has only slight impact on capturing more actual seeded errors.

We further examine overlaps between the actual seeded errors identified by LOF and K-Modes in each account class and find that, although LOF has a relatively poor performance, it still captures some seeded errors that are not identified by K-Modes. This finding illustrates the desirability of utilizing various outlier detection methods instead of relying on one method. By using both methods, auditors can attain a more comprehensive understanding of the data at hand.

V. EVALUATION OF MODF

Potential of the Framework

The Experiment section illustrates the effectiveness of the Multilevel Outlier Detection Framework (MODF) with a real-world and a synthetic GL dataset. In the experiment with the real-world GL data, the comparison of the risk assessment results based on the MODF and the auditors' weighted-risk-filter technique suggests that the MODF, as a full-population testing technique that has no mandatory requirement for auditors' judgement to identify unusual observations, can serve as a valuable tool for auditors to acquire new useful insights into GL data and clients, which may be transferred to future audit engagements. In contrast, in the experiment with the synthetic GL data, we create seeded errors, called dependency outliers, that are less likely, by their nature, to be identified by the weighted-risk-filter techniques to test the performance of the MODF to identify them. The accuracy rates suggest that the MODF can detect this type of outliers. Particularly, K-Modes is more suitable than LOF. Additionally, actual outliers identified by LOF and K-Modes do not fully overlap, which illustrates the importance of employing a variety of outlier detection methods in the MODF. This study provides a specific condition where the MODF outperforms the weighted-risk-filter technique.

Although the MODF was built in an audit context, it has the potential to be generalized to tax accounting or managerial accounting. For instance, the MODF can be utilized to detect tax refund fraud. We chose the external audit setting and the GL data to illustrate the MODF because managers typically have more detailed knowledge about the company than external auditors who need to identify material misstatements in a relatively short period from an overwhelmingly large number of financial transactions. Thus, outlier detection methods that require only limited knowledge about the data to identify unusual transactions would benefit external auditors more than other parties, such as internal auditors. Furthermore, the audit-related decision-making in each step allows the MODF to be easily customized to a specific audit engagement and to provide relevant outliers to auditors. For example, the thresholds in the prioritization step can be adjusted to a desired level of materiality.

Another advantage of using machine-learning-based outlier detection methods to identify unusual observations is that a client is less likely to be able to predict which transactions are considered normal (i.e., outlier scores are low) in an algorithm, making it more difficult to hide fraud. If a client produces fraudulent entries, attempts to mask them by making them similar to non-fraudulent entries will be complicated by the unpredictable and opaque mechanism used by an outlier detection method, increasing the chance for auditors to flag the fraud. As a result, the use of an outlier detection method could either identify fraudulent entries or deter clients from taking the risk to produce them. However, one drawback is that the client could produce many fraudulent entries that are similar to the non-fraudulent entries, which would increase the proportion of certain patterns in the population and make them more likely to be identified as normal. However, it is not a serious issue in most cases because those entries would be easily noticed by auditors.

Limitations of the Framework

Although the MODF aims to flag noticeable, audit-related observations in GL data, it may also produce false alerts, where are exceptional outliers that are shown to have no misstatement after examination by auditors. To mitigate this concern, auditors should apply the MODF to a client's prior engagement GL data and test whether it captures valuable observations. Additionally, future research can also explore internal evaluation of unsupervised outlier detection methods, which solely relies on data and output, the outlier scores, to evaluate a detector method. Internal evaluation will allow auditors to predict the performance of a detector method before examination.

Another limitation of the MODF methodology is that auditors may have difficulty in satisfying the requirement for audit documentation due to the machine-learning-based outlier detection methods employed. According to audit standards, audit documentation must enable an experienced auditor to understand "the procedures performed, evidence obtained, and conclusions reached (AS 1215.06, 2016)." However, the lack of interpretability of most machine learning methods poses a challenge for auditors to explain the reasoning underlying the method's outputs. Although research has proposed visualization and other techniques to understand how inputs are mathematically mapped to outputs in machine learning models, a general approach is still not yet available. Future research can focus on understanding the results of machine learning models, especially for unsupervised learning approaches like outlier detection. This will facilitate the inclusion of machine learning-based analytical methods, including this framework, into auditing procedures.

VI. CONCLUSION AND FUTURE RESEARCH

In this paper, we propose a framework called the Multilevel Outlier Detection Framework to explore risks in GL data. The framework involves three outlier detection processes to identify transaction-level, account-level, and CBV-level outliers. We demonstrate the framework using a real-world and a synthetic GL dataset. The investigation results indicate that outlier detection methods can be utilized to gain new insights regarding risk of material misstatements in the GL.

This paper contributes to both outlier detection and audit analytics literature by proposing a framework that systematically applies outlier detection methods to GL data. In addition, the two experiments in this paper provide evidence that auditors' knowledge and judgment may not always be sufficient to identify all risk factors related to errors and material misstatements.

The primary limitation of this paper is that the framework is tested with only two datasets. It is possible that new insights cannot always be acquired by examining the outliers from the MODF. Hence, we argue that it is more reasonable to position the framework as an additional defense against the risk of material misstatements. Specifically, if the internal controls are effective and the auditors have sufficient client-specific knowledge, then it is likely that the framework will find nothing new. Thus, auditors can issue their opinion report based on their existing risk assessment results with sufficient confidence. By contrast, if the internal controls are less effective and auditor's reaction to fraud cues is flawed, the MODF provides an additional way to search for abnormality in data, which may aid auditors to obtain new audit evidence.

Future research may test the framework with other data sources. It would also be interesting to understand how the application of such a framework affects auditor's judgment through behavioral research.

REFERENCES

Alawadhi, A. 2015. The application of data visualization in auditing. Doctoral dissertation, Rutgers, The State University of New Jersey, Newark. https://doi.org/doi:10.7282/T3GQ70MD.

Alghushairy, O., R. Alsini, T. Soule, and X. Ma. 2020. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing* 5(1):1. https://doi.org/10.3390/bdcc5010001.

American Institute of Certified Public Accountants (AICPA). 2017. *AICPA Guide to Audit Data Analytics*. Durham, NC: AICPA. Available at: https://us.aicpa.org/interestareas/frc/assuranceadvisoryservices/auditdataanalyticsguide.

Antweiler, W., and M. Z. Frank. 2004. Is all that talk just noise? The information content of Internet stock message boards. *The Journal of Finance* 59 (3): 1259-1293. https://doi.org/10.1111/j.1540-6261.2004.00662.x.

Asare, S. K., and A. M. Wright. 2004. The effectiveness of alternative risk assessment and program planning tools in a fraud setting. *Contemporary Accounting Research* 21 (2): 325–52. https://doi.org/10.1506/L20L-7FUM-FPCB-7BE2.

Beck, P. J., and I. Solomon. 1985. Sampling risks and audit consequences under alternative testing approaches. *The Accounting Review* 60 (4): 714-723. https://www.jstor.org/stable/247467.

Blocher, E., and J. H. Bylinski. 1985. The influence of sample characteristics in sample evaluation. *Auditing: A Journal of Practice & Theory* 5 (1): 79-90.

Breunig, M. M., H. P. Kriegel, R. T. Ng, and J. Sander. 2000. LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93-104. https://doi.org/10.1145/342009.335388.

Cynthia, P. C., and S. T. George. 2021. An outlier detection approach on credit card fraud detection using machine learning: A comparative analysis on supervised and unsupervised learning. In *Intelligence in Big Data Technologies—Beyond the Hype*, edited by J. D. Peter, S. L. Fernandes, and A. H. Alavi, 1167:125–135. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore. https://doi.org/10.1007/978-981-15-5285-4_12.

Davis, J. J., and A. J. Clark. 2011. Data preprocessing for anomaly based network intrusion detection: A review. *Computers & Security* 30 (6–7): 353–375. https://doi.org/10.1016/j.cose.2011.05.008.

Debreceny, R. S., and G. L. Gray. 2010. Data mining journal entries for fraud detection: An exploratory study. *International Journal of Accounting Information Systems* 11 (3): 157–181. https://doi.org/10.1016/j.accinf.2010.08.001.

Dilla, W., D. J. Janvrin, and R. Raschke. 2010. Interactive data visualization: New directions for accounting information systems research. *Journal of Information Systems* 24 (2): 1–37. https://doi.org/10.2308/jis.2010.24.2.1.

Elder, R. J., and R. D. Allen. 1998. An empirical investigation of the auditor's decision to project errors. *Auditing: A Journal of Practice & Theory* 17(2): 71-87.

Freiman, J. W., Y. Kim, and M. A. Vasarhelyi. 2022. Full population testing: Applying multidimensional audit data sampling (MADS) to general ledger data auditing. *International Journal of Accounting Information Systems* 46: 100573. https://doi.org/10.1016/j.accinf.2022.100573.

Grubbs, F. E. 1969. Procedures for detecting outlying observations in samples. *Technometrics* 11 (1): 1–21. https://doi.org/10.1080/00401706.1969.10490657.

Hall, T. W., J. E. Hunton, and B. J. Pierce. 2000. The use of and selection biases associated with nonstatistical sampling in auditing. *Behavioral Research in Accounting* 12: 231-255.

Hoffman, V. B., and M. F. Zimbelman. 2009. Do strategic reasoning and brainstorming help auditors change their standard audit procedures in response to fraud risk? *The Accounting Review* 84 (3): 811–37. https://doi.org/10.2308/accr.2009.84.3.811.

Huang, Z. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery* 2: 283-304. https://doi.org/10.1023/A:1009769707641.

Issa, H. 2013. Exceptional exceptions. Doctoral dissertation, Rutgers, The State University of New Jersey, Newark. https://doi.org/doi:10.7282/T32J68V1.

Issa, H., and A. Kogan. 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.

Jans, M., M. Alles, and M. A. Vasarhelyi. 2013. The case for process mining in auditing: Sources of value added and areas of application. *International Journal of Accounting Information Systems* 14 (1): 1–20. https://doi.org/10.1016/j.accinf.2012.06.015.

Khan, R., A. Clark, G. Mohay, and S. Suriadi. 2014. Detecting fraud using transaction frequency data. *Information Technology in Industry* 2 (3). Available at: http://www.it-in-industry.org/index.php/itii/article/view/18.

Khan, R., M. Corney, A. Clark, and G. Mohay. 2010. Transaction mining for fraud detection in ERP systems. *Industrial Engineering and Management Systems* 9 (2): 141–156. https://doi.org/10.7232/iems.2010.9.2.141.

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

Liu, Q. 2014. The application of exploratory data analysis in auditing. Doctoral dissertation, Rutgers, The State University of New Jersey, Newark. https://doi.org/doi:10.7282/T3CC129J.

Loughran T., and B. McDonald. 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance* 66 (1): 35-65. https://doi.org/10.1111/j.1540-6261.2010.01625.x.

Mokua, N., C. W. Maina, and H. Kiragu. 2021. Anomaly detection for raw water quality – A comparative analysis of the local outlier factor algorithm and the random forest algorithms. *International Journal of Computer Applications* 174 (26): 47–54. https://doi.org/10.5120/ijca2021921196.

No, W. G., K. Lee, F. Huang, and Q. Li. 2019. Multidimensional audit data selection (MADS): A framework for using data analytics in the audit data selection process. *Accounting Horizons* 33 (3): 127–140. https://doi.org/10.2308/acch-52453.

Public Company Accounting Oversight Board (PCAOB). 2016. *Auditing Standards* (AS 1215.06). Retrieved from https://pcaobus.org/Standards/Auditing/Pages/ReorgStandards.aspx.

Public Company Accounting Oversight Board (PCAOB). 2016. *Auditing Standards* (AS 2110.46). Retrieved from https://pcaobus.org/Standards/Auditing/Pages/ReorgStandards.aspx.

Ramaswamy, S., R. Rastogi, and K. Shim. 2000. Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record* 29 (2): 427-438. https://doi.org/10.1145/335191.335437.

Srivastava, A., A. Kundu, S. Sural, and A. K. Majumdar. 2008. Credit card fraud detection using hidden Markov model. *IEEE Transactions on Dependable and Secure Computing* 5 (1): 37–48. https://doi.org/10.1109/TDSC.2007.70228.

Steinbuss, G., and K. Böhm. 2021. Benchmarking unsupervised outlier detection with realistic synthetic data. *ACM Transactions on Knowledge Discovery from Data* 15 (4): 1–20. https://doi.org/10.1145/3441453.

Teitlebaum, A. D., and C. F. Robinson. 1975. The real risks in audit sampling. *Journal of Accounting Research* 13: 70–91. https://doi.org/10.2307/2490480.

Tetlock, P. C. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance* 62 (3): 1139-1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x.

Thiprungsri, S., and M. Vasarhelyi. 2011. Cluster analysis for anomaly detection in accounting data: An audit approach. *The International Journal of Digital Accounting Research* 11. https://doi.org/10.4192/1577-8517-v11_4.

Tukey, J. W. 1977. *Exploratory Data Analysis*. Reading, Massachusetts: Addison-Wesley.

Wilks, T. J., and M. F. Zimbelman. 2004. Using game theory and strategic reasoning concepts to prevent and detect fraud. *Accounting Horizons* 18 (3): 173–84. https://doi.org/10.2308/acch.2004.18.3.173.

Zhaokai, Y., and K. C. Moffitt. 2019. Contract analytics in auditing. *Accounting Horizons* 33 (3): 111–126. https://doi.org/10.2308/acch-52457.

Zhuang, X., Y. Huang, K. Palaniappan, and Y. Zhao. 1996. Gaussian mixture density modeling, decomposition, and applications. *IEEE Transactions on Image Processing* 5 (9): 1293–1302. https://doi.org/10.1109/83.535841.

Figure 1. Multilevel Outlier Detection Framework

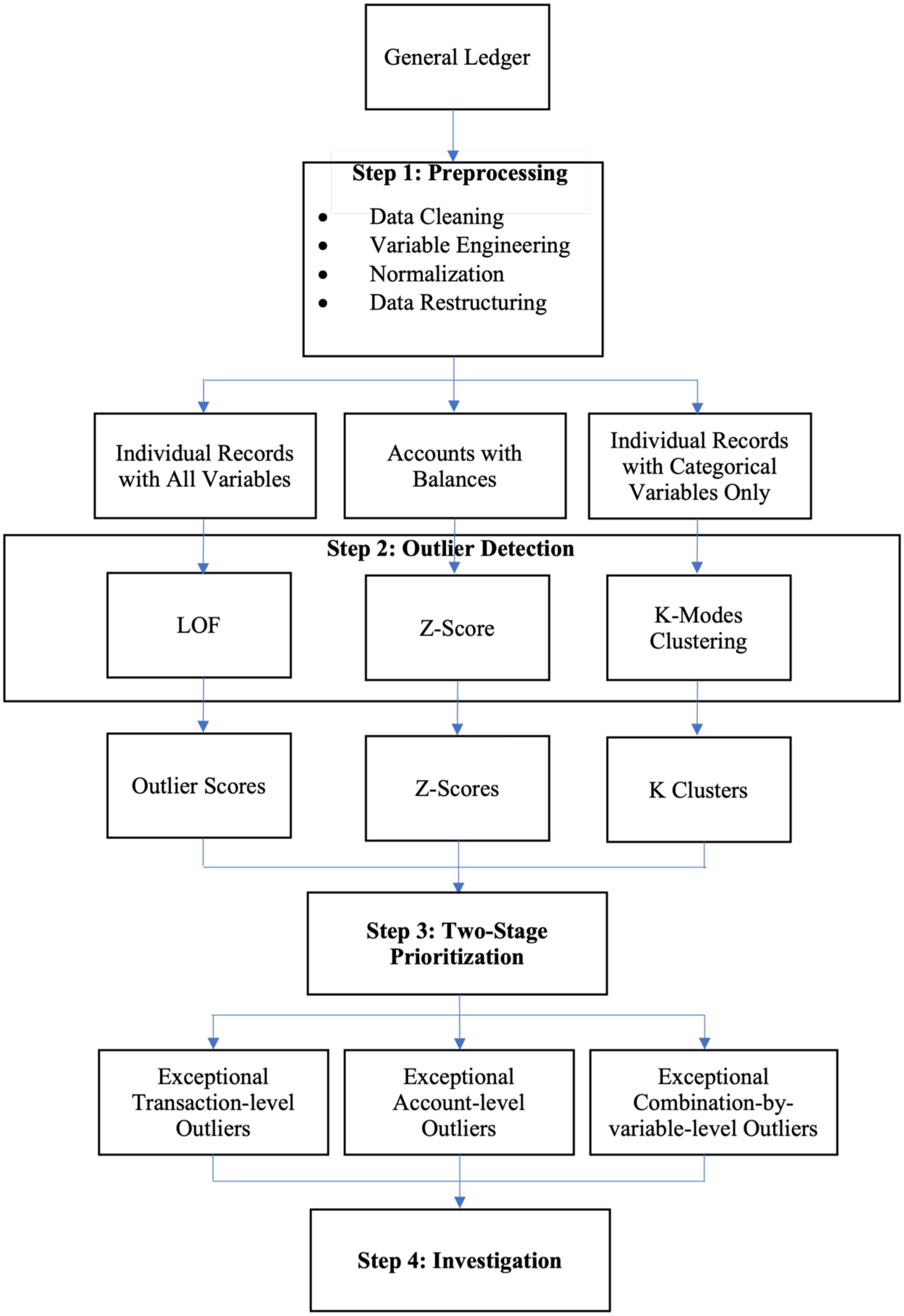
General Ledger
Step 1: Preprocessing
Data Cleaning
Variable Engineering
Normalization
Data Restructuring
Individual Records with All Variables
Accounts with Balances
Individual Records with Categorical Variables Only
Step 2: Outlier Detection
LOF
Z-Score
K-Modes Clustering
Outlier Scores
Z-Scores
K Clusters
Step 3: Two-Stage Prioritization
Exceptional Transaction-level Outliers
Exceptional Account-level Outliers
Exceptional Combination-by-variable-level Outliers
Step 4: Investigation

Table 1. Variable Descriptions – Real Data

Variable Name	*Type*	*Description*
Account Code	Categorical	General ledger account ID (corresponding to the "Account Number" in Trial Balance)
Transaction ID	Categorical	Journal entry transaction ID (not unique)
Net	Numerical	Net debit/ credit amount posted by the journal line
Effective Date	Date	Date the transaction occurred
Created Date/ Time	Date	Date the transaction was processed into the system
Document Type	Categorical	Type of transaction and source that was posted
User ID	Categorical	Individual or System ID that entered the transaction
Journal Description	Textual	Narration for the journal
Line Description	Textual	Specific narration of the line of the journal

Table 2. Summary of Variable Preprocessing – Real Data

Variable Name	*Description*
Account Code	Converted into binary variables by one-hot encoding.
Transaction ID	Dropped because the information can be derived from the other variables.
Net	The absolute values are normalized.
Effective Date	Replaced by a new variable called Day Difference.
Created Date/ Time	Replaced by a new variable called Day Difference.
Document Type	Converted into binary variables by one-hot encoding.
User ID	Converted into binary variables by one-hot encoding.
Journal Description	Replaced by a new variable called Keyword Count.
Line Description	Dropped because all records have missing values.

Table 3. Number of Records, Accounts, and Unique Variable Combinations in Six GL Categories – Real Data

GL Category	*Index*	*# of Records*	*# of Accounts*	*# of Unique Variable Combinations*
Cash and Cash Equivalents	A	6,034	13	56
Creditors, Accruals, and Settlement Accounts	B	1,341	24	80
Depreciation and Amortization	C	2,577	18	72
Derivative Assets Held for Hedging Purposes	D	40,285	10	20
Employee Compensation and Benefits	E	10,198	193	1,256
Fee, Commission, and Other Income	F	60,743	158	733
Total	NA	121,178	416	2,217

Table 4. Two-Stage Prioritization Results – Real Data

Panel A. Transaction Level

GL Category Index	*# of Records*	*# of Notable Records by the 75th Percentile Threshold*	*Stage-One Threshold Applied*	*Stage-Two Threshold Applied*	*# of Exceptional Records*
A	6,034	1,508	Top 100	$10,000	10
B	1,341	335	Top 100	$10,000	22
C	2,577	644	Top 100	$10,000	1
D	40,285	10,071	Top 100	$10,000	8
E	10,198	2,549	Top 100	$10,000	0
F	60,743	15,185	Top 100	$10,000	0
Total	**121,178**	**30,292**	**NA**	**NA**	**41**

Panel B. Account Level

GL Category Index	*# of Accounts*	*# of Notable Accounts by the 75th Percentile Threshold*	*Stage-One Threshold Applied*	*Stage-Two Threshold Applied*	*# of Exceptional Accounts*
A	13	3	75th Percentile	$10,000	2
B	24	6	75th Percentile	$10,000	6
C	18	5	75th Percentile	$10,000	2
D	10	3	75th Percentile	$10,000	3
E	193	48	75th Percentile	$10,000	18
F	158	40	75th Percentile	$10,000	19
Total	**416**	**105**	**NA**	**NA**	**50**

Panel C. Combination-by-Variable Level

GL Category Index	*# of Clusters*	*# of Notable Unique Combos by the 75th Percentile Threshold*	*Stage-One Threshold Applied*	*Stage-Two Threshold Applied*	*# of Unique Exceptional Combos*
A	20	47	75th Percentile	$10,000	11
B	40	47	75th Percentile	$10,000	13
C	25	41	75th Percentile	$10,000	8
D	4	3	75th Percentile	$10,000	1
E	900	699	Top 100	$10,000	1
F	500	458	Top 100	$10,000	9
Total	**1,489**	**1,295**	**NA**	**NA**	**43**

Note: For each level of observations, if the number of notable outliers based on the 75th percentile threshold does not exceed 100, the 75th percentile threshold is applied. Otherwise, the top-100 threshold will override. The threshold used is listed in the "Stage-One Threshold Applied" column. The stage-two threshold is the same for all the three levels, which is $10,000. The last column lists the number of exceptional outliers at each level.

Table 5. MODF Risk Assessment Results of Exceptional Records, Accounts, and Combinations – Real Data

Panel A. Risk Levels of Exceptional Records

GL Category Index	*# Of Exceptional Records*	*Low Risk*	*Medium Risk*	*High Risk*
A	10	7 (70%)	1 (10%)	2 (20%)
B	22	19 (86.4%)	3 (13.6%)	0 (0%)
C	1	0 (0%)	1 (100%)	0 (0%)
D	8	8 (100%)	0 (0%)	0 (0%)
E	0	0 (0%)	0 (0%)	0 (0%)
F	0	0 (0%)	0 (0%)	0 (0%)
Total	**41**	**34 (82.9%)**	**5 (12.2%)**	**2 (4.9%)**

Panel B. Risk Levels of Exceptional Accounts

GL Category Index	*# of Exceptional Accounts*	*Low Risk*	*Medium Risk*	*High Risk*
A	2	2 (100%)	0 (0%)	0 (0%)
B	6	4 (66.7%)	2 (33.3%)	0 (0%)
C	2	1 (50%)	1 (50%)	0 (0%)
D	3	2 (66.7%)	1 (33.3%)	0 (0%)
E	18	0 (0%)	18 (100%)	0 (0%)
F	19	0 (0%)	19 (100%)	0 (0%)
Total	**50**	**9 (18%)**	**41 (82%)**	**0 (0%)**

Panel C. Risk Levels of Exceptional Combinations

GL Category Index	*# Of Exceptional Combinations*	*Low Risk*	*Medium Risk*	*High Risk*
A	11	6 (54.5%)	5 (45.5%)	0 (0%)
B	13	5 (38.5%)	5 (38.5%)	3 (23.0%)
C	8	1 (12.5%)	7 (87.5%)	0 (0%)
D	1	1 (100%)	0	0
E	1	0 (23.1%)	1 (100%)	0 (0%)
F	9	0 (0%)	4 (44.4%)	5 (55.6%)
Total	**43**	**13 (30.2%)**	**22 (51.2%)**	**8 (18.6%)**

Table 6. Comparison of Risk Assessment Results Between MODF and Weighted-Risk-Filter Technique

Panel A. Exceptional Records	
Number of Exceptional Records	***Number of Exceptional Records with Higher Newly Assigned Risk***
41	0 (0%)
Panel B. Exceptional Accounts	
Number of Exceptional Accounts	***Number of Exceptional Accounts with Higher Newly Assigned Risk***
50	5 (10%)
Panel C. Exceptional Combinations	
Number of Exceptional Combinations	***Number of Exceptional Combinations with Higher Newly Assigned Risk***
43	29 (67%)

Table 7. Variable Descriptions - Synthetic Data

Variable Name	*Type*	*Description*
Account Class	Categorical	Specific classification of account (e.g., payroll expenses).
Account Type	Categorical	Type of account (e.g., asset, liability).
Amount	Numerical	Total amount of the journal entry line item (may be positive or negative).
Business Unit	Categorical	The business unit (e.g., hotel, food and beverage) of the journal entry.
Credit	Numerical	Credit amount of the entry.
Debit	Numerical	Debit amount of the entry.
Effective Date	Date	Date the entry was posted to the GL as occurring.
Entry Date	Date	Date that the entry was entered into the subsystem or GL, depending on the type of transaction.
GL Account Name	Categorical	Name of the GL account.
GL Account Number	Categorical	GL account number.
JE Description	Textual	Description of the transaction. May include vendor or guest name, etc.
JE Identifier	Categorical	Unique identifier for each journal entry.
Period	Date	Indicates which month within the fiscal year the transaction occurred.
Preparer ID	Categorical	The employee ID for the employee who initiated the transaction.
Source	Categorical	Describes the payment type or other source type of the transaction (e.g., CASH RECEIPT).

Table 8. Summary of Variable Preprocessing - Synthetic Data

Variable Name	*Description*
Account Class	Dropped because all records in an account class have the same value.
Account Type	All records have the same value as the accounts are separated based on Account Class.
Amount	The absolute values are normalized.
Business Unit	Converted into binary variables by one-hot encoding.
Credit	Dropped because it has a 1-to-1 relationship with Amount.
Debit	Dropped because it has a 1-to-1 relationship with Amount.
Effective Date	Replaced by a new variable called Day Difference.
Entry Date	Replaced by a new variable called Day Difference.
GL Account Name	It has a 1-to-1 relationship with GL Account Number.
GL Account Number	Converted into binary variables by one-hot encoding.
JE Description	Dropped because a keyword dictionary is not available for variable engineering.
JE Identifier	Dropped because the information can be derived from the other variables.
Period	Dropped because the information can be derived from the other variables.
Preparer ID	Converted into binary variables by one-hot encoding.
Source	Converted into binary variables by one-hot encoding.

Table 9. Number of Records and Unique Variable Combinations in Seven Account Classes (Original and with Seeded Errors) – Synthetic Data

Account Class	*# of Records (Original)*	*# of Accounts (Original)*	*# of Unique Combinations (Original)*	*Seeded Error Rate 1%*		*Seeded Error Rate 3%*		*Seeded Error Rate 5%*	
				# of Records	*# of Unique Combinations*	*# of Records*	*# of Unique Combinations*	*# of Records*	*# of Unique Combinations*
Accrued Payroll	2,064	1	18	2,084	29	2,127	39	2,172	41
Cash	16,768	1	74	16,937	192	17,286	310	17,650	373
Inventory	2,869	2	14	2,897	25	2,957	32	3,020	35
SG&A	4,027	38	197	4,067	236	4,151	303	4,238	368
Sales-Dining	2,881	1	7	2,910	13	2,970	14	3,032	14
Sales-Hotel	3,892	1	13	3,931	18	4,012	23	4,096	26
Sales-Other	1,756	3	13	1,773	25	1,810	34	1,848	40
Total	**34,257**	**47**	**336**	**34,599**	**538**	**35,313**	**755**	**36,056**	**897**

Note: The table lists the numbers of records, accounts, and unique combinations in the original dataset, and the new numbers of records, and unique combinations after 1-, 3-, and 5-percent seeded errors added to the dataset. The number of accounts does not change with seeded errors because no new account is generated. In the account classes where the number of accounts is one, "GL Account Number" is not used for outlier detection because all the records have the same value.

Table 10. Outlier Detection Results – Synthetic Data

Panel A. Accuracy Rate of LOF with Top 100 and 200 Threshold

	Seeded Error Rate 1%		***Seeded Error Rate 3%***		***Seeded Error Rate 5%***	
Account Class	***Accurate Rate- Top 100***	***Accurate Rate- Top 200***	***Accurate Rate- Top 100***	***Accurate Rate- Top 200***	***Accurate Rate- Top 100***	***Accurate Rate- Top 200***
Accrued Payroll	0.18 **(0.20)**	0.09 **(0.10)**	0.34 **(0.63)**	0.28 **(0.32)**	0.38 **(1.00)**	0.36 **(0.54)**
Cash	0.13 **(1.00)**	0.11 **(0.85)**	0.30 **(1.00)**	0.21 **(1.00)**	0.42 **(1.00)**	0.33 **(1.00)**
Inventory	0.17 **(0.28)**	0.11 **(0.14)**	0.44 **(0.88)**	0.29 **(0.44)**	0.56 **(1.00)**	0.41 **(0.76)**
SG&A	0.14 **(0.40)**	0.11 **(0.20)**	0.32 **(1.00)**	0.23 **(0.62)**	0.41 **(1.00)**	0.31 **(1.00)**
Sales-Dining	0.22 **(0.29)**	0.13 **(0.15)**	0.47 **(0.89)**	0.27 **(0.45)**	0.63 **(1.00)**	0.43 **(0.76)**
Sales-Hotel	0.07 **(0.39)**	0.04 **(0.20)**	0.13 **(1.00)**	0.07 **(0.60)**	0.19 **(1.00)**	0.10 **(1.00)**
Sales-Other	0.13 **(0.17)**	0.09 **(0.09)**	0.22 **(0.54)**	0.19 **(0.27)**	0.23 **(0.92)**	0.22 **(0.46)**

Panel B. Accuracy Rate of K-Modes with Top 100 and 200 Threshold

	Seeded Error Rate 1%		***Seeded Error Rate 3%***		***Seeded Error Rate 5%***	
Account Class	***Accurate Rate- Top 100***	***Accurate Rate- Top 200***	***Accurate Rate- Top 100***	***Accurate Rate- Top 200***	***Accurate Rate- Top 100***	***Accurate Rate- Top 200***
Accrued Payroll	0.20 **(0.20)**	0.10 **(0.10)**	0.63 **(0.63)**	0.32 **(0.32)**	0.66 **(1.00)**	0.54 **(0.54)**
Cash	0.64 **(1.00)**	0.38 **(0.85)**	0.92 **(1.00)**	0.76 **(1.00)**	0.93 **(1.00)**	0.86 **(1.00)**
Inventory	0.24 **(0.28)**	0.12 **(0.14)**	0.63 **(0.88)**	0.35 **(0.44)**	0.74 **(1.00)**	0.60 **(0.76)**
SG&A	0.34 **(0.40)**	0.17 **(0.20)**	0.70 **(1.00)**	0.51 **(0.62)**	0.76 **(1.00)**	0.75 **(1.00)**
Sales-Dining	0.23 **(0.29)**	0.12 **(0.15)**	0.66 **(0.89)**	0.33 **(0.45)**	0.96 **(1.00)**	0.57 **(0.76)**
Sales-Hotel	0.32 **(0.39)**	0.17 **(0.20)**	0.65 **(1.00)**	0.47 **(0.60)**	0.74 **(1.00)**	0.77 **(1.00)**
Sales-Other	0.15 **(0.17)**	0.09 **(0.09)**	0.51 **(0.54)**	0.27 **(0.27)**	0.80 **(0.92)**	0.45 **(0.46)**

Note: The table shows the accuracy rates of LOF and K-Modes with the top-100 and top-200 thresholds, which are calculated as the number of actual seeded errors identified among the top 100 (200) observations divided by 100 (200). The rates in bold are the possible highest rates with the top-100 and top-200 thresholds, which are calculated by the total number of actual seeded errors divided by 100 (200). If the number is over 100 (200), the possible highest rate is 1.0.

Appendix A. Example of Journal Entry Data

GL Category	*Account Code*	*Transaction ID*	*Net*	*Effective Date*	*Created Date / Time*	*Document Type*	*User ID*	*Journal Description*	*Line Description*
Cash and Cash Equivalent	10.1011. 30944	PCARD	88	6/6/18	17/09/18	PCSTAT	User 1	Cyber Resilience	
Employee Compensation and Benefits	10.3000. 60408	46174	10182.56	18/12/18	19/12/18	APJNL	User 2	Staff Pays WE	
Derivative Assets Held for Hedging Purposes	10.1014. 10220	42860	-11.18	24/7/18	01/08/18	APJNL	User 3	GST ADJ JUN 18 QTR BAS	

Note: The three entries are selected from different GL categories to illustrate of the entry data used in the experiment. Variable values include confidential information have been removed.