CONTINUOUS AUDIT ANALYTICS METHODS: 
THE SKIPPER, THE STRETCHER, AND THE LOOPER.

by

Nuriddin Tojiboyev

A dissertation submitted to the

Graduate School – Newark

Rutgers, The State University of New Jersey

In partial fulfillment of requirements

for the degree of

Doctor of Philosophy

Graduate Program in Management

written under the direction of

Alexander Kogan

and approved by


Newark, New Jersey

May, 2022
ABSTRACT OF THE DISSERTATION

Continuous Audit Analytics Methods:
the Skipper, the Stretcher, and the Looper

By Nuriddin Tojiboyev

Dissertation Director: Dr. Alexander Kogan

The transaction-level implementation of a continuous auditing framework requires the analysis of exceptions and anomalies. Scholars used various data analytics tools to contribute to the evolution of the framework. This doctoral dissertation advances the existing literature with different risk aggregation methods, similarity-aware examination, and a confirmed instance-based feedback loop. The first chapter identifies the “risk position” of a transaction in the “risk space” and uses different distance metrics to calculate the aggregate risk score for the transaction. While variations of Minkowski distance allow users to balance the risk calculation from multiple filter values, the utilization of Mahalanobis distance considers the relationships among filters and adjusts for correlations during the aggregation of filter values.

In the second chapter, I propose using similarity constraints during the audit selection procedures to avoid selecting accounting items that represent similar risks which may cause suboptimal use of some audit resources. I introduce two relevant objectives to the audit analytics literature: risk maximization and similarity minimization. Following these objectives, I develop two algorithms: the Skipper and the Stretcher, each prioritizing one of the objectives while holding the other objective bound by a user-defined threshold.
In the third chapter, I contribute a feedback loop method to the existing literature on continuous audit that learns and updates the boundaries of the high-risk subspaces by analyzing the confirmed instances from the investigation of exceptions and outliers. Additionally, this feedback loop also recommends filters that might be missing from the transaction verification module. To evaluate the model, I run a simulation of a multiperiod continuous audit framework with a proposed feedback loop model, learning and adjusting the filters of the transaction verification throughout nine periods of data. This model allows auditors to update filter thresholds, add new filters or remove filters that became obsolete.
ACKNOWLEDGEMENTS

I had the greatest pleasure of working with and under the supervision of wonderful people during my PhD years at Rutgers University. For the past several years, I have been blessed to be around the great faculty members of the Accounting and Information Systems Department and have had a great opportunity of learning from them.

First and foremost, I thank Professor Alexander Kogan for his guidance and mentoring in my academic and personal life. The wisdom and knowledge he shared were essential in my development as an educator and researcher. Without his continuous support and motivation, I would not have made it to these days of my life. I am also grateful to Professor Miklos Vasarhelyi for his academic nurturement through projects and discussions during my PhD studies. His faith in my success was a strong motivator to pursue academic achievements. The research projects he involved me served as foundations to the chapters of this doctoral dissertation.

I want to express my deepest appreciation to Professors Won Gyun No and Deniz Appelbaum for their valuable comments and feedback, the department chair Professor Dan Palmon for his support and advice on my teaching, and Barbara Jensen for her efforts to make our academic journey unforgettable. Additionally, I thank my friends, Cohort of 2017 (AIS Concentration), Dr. Eid Alotaibi, Dr. Areon Cheong, Dr. Sheng-Feng Hsieh, and always glamourous Chanyuan (Abby) Zhang-Parker.

Finally, I thank my family, each of whom I love with my heart and soul.

Praise be to God, the Lord of the Worlds!
# TABLE OF CONTENTS

ABSTRACT OF THE DISSERTATION ................................................................. ii

ACKNOWLEDGEMENTS ................................................................................ iv

LIST OF TABLES .......................................................................................... x

INTRODUCTION ............................................................................................ 1

CHAPTER 1: DISTANCES AND ANGLES OF RISK SCORE ....................... 1
  Literature Review ....................................................................................... 1
  Risk Composition and Risk Space ............................................................. 3
  Distance for Aggregation .......................................................................... 6
  Evaluation of Distances for Risk Scores: Case Study .............................. 12
  Summary .................................................................................................. 24

CHAPTER 2: SKIPPER AND STRETCHER ALGORITHMS FOR AUDIT
  SELECTION ................................................................................................. 25
  Motivation for Research .......................................................................... 26
  Use of Similarity in Selection .................................................................. 28
  The Skipper ............................................................................................. 34
  The Stretcher .......................................................................................... 36
  Evaluation of Algorithms: Application on Synthetic Data ...................... 40
  Evaluation of Algorithms: Application on Real-Life Data .................... 45
  Modification for Special Cases ............................................................... 52
Summary.......................................................................................................................... 52

ESSAY THREE: EXCEPTION RULE GENERATION FROM OUTLIERS ........ 54

Motivation for Research ........................................................................................................ 55

Causes of Uncaught Exceptions .......................................................................................... 59

Rule Induction from Confirmed Outliers ............................................................................ 62

Evolution of Rules in Multiperiod Implementation ............................................................ 67

Evaluations of the Model ....................................................................................................... 70

Factors Affecting the Performance of the Model ................................................................. 76

Summary ................................................................................................................................ 78

CONCLUSIONS ...................................................................................................................... 80

REFERENCES ......................................................................................................................... 84

APPENDIX A: DATA DICTIONARY FOR A PAYROLL DATASET ......................... 87

APPENDIX B: HOW VARIANCES AND COVARIANCES AFFECT MAHALANOBIS DISTANCE ................................................................. 88

APPENDIX C: DECOMPOSITION OF COVARIANCE MATRIX ............................ 91

APPENDIX D: GREEDY HEURISTIC APPROACH BASED A RECURSIVE RULE INDUCTION ......................................................................................................................... 92
LIST OF FIGURES

Figure 1. Risk space concept (left) and distance as risk score (right)......................... 6
Figure 2. Tolerable risk region of the risk space. .......................................................... 7
Figure 3. Conversion of covariance matrix into an inverse matrix................................. 11
Figure 4. Horizontal variance differences of the exception pairs of relations that change under Euclidean and Mahalanobis distances ................................................................. 19
Figure 5. The intersections portions for different selections sizes under Manhattan and Euclidean-based risk scores ........................................................................................................ 20
Figure 6. The intersections portions for different selections sizes under on Euclidean and Mahalanobis -based risk scores ........................................................................................................ 23
Figure 7. Risk Composition Similarity among Exceptions Sorted by Risk Score............. 27
Figure 8. Scatter Plot of Data Points of Records by Relevant Data Fields.................... 29
Figure 9. Scatter Plot of Record Risk Filter Values in Risk Space............................... 31
Figure 10. Visualization of the Skipper Execution on a Sorted List of Exceptions. ....... 34
Figure 11. Steps of the Stretcher Algorithm .................................................................... 37
Figure 12. Visualization of the Stretcher Execution on a Sorted List of Exceptions. ..... 37
Figure 13. The Scatter Plot of Synthetic Datasets Used for Evaluation ....................... 40
Figure 14. The Scatter Plot of Selected Exceptions from the Synthetic Datasets without Use of Similarity .................................................................................................................. 41
Figure 15. The Scatter Plot of Selected Exceptions from the Synthetic Datasets using the Skipper Algorithm. ............................................................................................................ 42
Figure 16. The Scatter Plot of Selected Exceptions from the Synthetic Datasets using the Stretcher Algorithm. ........................................................................................................... 44
Figure 17. The Histogram of Exceptions in Payroll Dataset by Euclidean Distance to the Closest Exception................................................................. 46

Figure 18. Relationship between Similarity Threshold (Min_Distance) and Minimum Risk Score of Selected Exceptions from the Payroll Dataset using the Skipper Algorithm. ......................................................................................... 48

Figure 19. Relationship between Risk Score Threshold (Min_Risk_Score) and Similarity (Distance) of Selected Exceptions from the Payroll Dataset using the Stretcher Algorithm.......................................................... 51

Figure 20. The architecture of the Continuous Data Level Auditing System (adapted from Kogan, Alles, Vasarhelyi, and Wu, 2014) ................................................................. 56

Figure 21. Feedback to Transaction Verification from Confirmed Misstatements of Anomalies and Outliers........................................................................ 57

Figure 22. Causes of undetected exceptions by transaction verification.................. 60

Figure 23. Input data for rule induction from confirmed outliers........................... 63

Figure 24. Moving window approach for data input to rule induction algorithms....... 64

Figure 25. Interactions of Maximum Rule Complexity, Maximum Negative Class Ratio, and Minimum Positive Instances parameters. ................................................... 67

Figure 26. Merging of rules due to a rule generalization........................................ 69

Figure 27. Changes in filter threshold over time. ..................................................... 74

Figure 28. The portion of the errors caught and the false positive rate of exception reported by the model. ................................................................. 76

Figure 29. Inversion of a diagonal matrix.............................................................. 89

Figure 30. Inversion of a diagonal matrix.............................................................. 89
Figure 31. Separate effects of variance and covariance on Mahalanobis Distance .......... 90
Figure 32. Decomposition of Mahalanobis Distance to variance and covariance effects. 91
Figure 33. First Rule Induction Run for Period 1. ................................................. 92
Figure 34. Second Rule Induction Run for Period 1............................................. 92
Figure 35. Third Rule Induction Run for Period 1................................................. 93
LIST OF TABLES

Table 1. Example of filter values for different transactions. ........................................... 4

Table 2. Filters and their formulas to calculate filter values of transactions. .................... 12

Table 3. The descriptive statistics on unweighted filter values of payroll data records. .. 13

Table 4. Excerpt from the list of exceptions with the highest risk. ................................. 14

Table 5. Correlation matrix of filter values. ....................................................................... 14

Table 6. Descriptive statistics on the normalized filter values. ......................................... 15

Table 7. An excerpt from the relation matrix for the exceptions that are ordered using

Manhattan distance. ........................................................................................................... 16

Table 8. Exception relation counts by their relation values under Manhattan and

Euclidean distances. ......................................................................................................... 17

Table 9. Exception relation counts by their relation values under Euclidean and

Mahalanobis distances. .................................................................................................... 21

Table 10. List of Exceptions that are equivalent in their risk scores under Manhattan and

Euclidean but not under Mahalanobis. .............................................................................. 22

Table 11. Formalization of the similarity and risk score based optimization.................... 33

Table 12. Descriptive Statistics of Datapoints of the Synthetic Datasets. ......................... 41

Table 13. The Descriptive Statistics on Distances to Closest Selected Neighbor of

Selected Exceptions from the Synthetic Datasets using the Skipper Algorithm. .......... 42

Table 14. The Descriptive Statistics on Distances to Closest Selected Neighbor of

Selected Exceptions from the Synthetic Datasets using the Stretcher Algorithm. ...... 44

Table 15. Descriptive Statistics of Selection Exceptions from Payroll Dataset Using the

Skipper Algorithm with Different Dissimilarity Thresholds........................................ 47
Table 16. Distribution of Exceptions in the Payroll Dataset by Risk Scores. .................. 49

Table 17. Descriptive Statistics of Selection Exceptions from Payroll Dataset Using the Stretcher Algorithm with Different Risk Score Thresholds. .............................................. 50

Table 18. Filters, rules, and rule thresholds obtained from applying the recursive CART on a payroll dataset. ........................................................................................................ 73

Table 19. Accuracy of filters on predicting error of following period. ............................. 75

Table 20. Information about the fields of the payroll dataset used for the evaluation of the models. ......................................................................................................................... 87
INTRODUCTION

The continuous auditing framework (Vasarhelyi and Halper 1991) has been expanding and evolving for 30 years. The utilization of diverse analytical methods contributed to the evolution and expansion of the framework applied in different audit settings. While initial concepts (Vasarhelyi and Halper, 1991; Alles et al., 2006; Alles et al., 2008) were based on rule-based auditing of a population of records and transactions, later additions to the framework (Kogan et al., 2014; Issa and Kogan 2014; Li et al., 2016; No et al., 2019) included the incorporation of analyses of outliers and anomalies. The generally accepted framework (Vasarhelyi and Halper 1991) utilizes tests for known audit risks through filters derived from business rules and internal controls. The unknown audit risks are assessed and managed through analyses of outliers and anomalies at aggregate levels (Kogan et al., 2014). The effectiveness and interoperability of these two components are critical to the successful implementation of a continuous audit framework.

In the first chapter, I explore the importance of the aggregation method for rule-based risk filters. In continuous auditing, a risk filter can be developed for each risk auditor is aware. Each risk filter assesses the riskiness of each item in the population from a different audit risk perspective. To understand the risks that the auditor is aware of, I propose a multidimensional space whose axes represent levels of different risks. The number of different risk filters sets the number of dimensions in this “risk space”. Since the origin of space is a zero risk point for all filters, it is also a risk-free point of the overall risk space. Distance from the origin to the position of any transaction or a record in this “risk space” can be used as a metric for an aggregated measure of known risks. I evaluate
the implication of using different distance metrics (Manhattan, Euclidean, Minkowski, Mahalanobis)\(^1\) as a risk score aggregation tool.

The published literature that proposes continuous audit models (Vasarhelyi and Halper, 1991; Kogan et al., 2014; Issa and Kogan, 2014; Li et al., 2016; No et al., 2019) does not discuss the implications of relationships among filter values and their effect on risk scores. Even with the efforts to keep rule-based filters mutually exclusive by allowing them to assess unrelated audit risk types, I have found that some recordkeeping practices of accounting systems do not allow absolute exclusiveness among filters\(^2\). In the cases of filters assessing the same risk factors, it would be inappropriate to take in values of these filters to the risk score without adjustments or discounts, as it would result in an over-assessment of risk by risk scores. One of the proposed models in this dissertation would address this problem by using Mahalanobis distance as a risk score measure.

In the second chapter, I explain the importance of introducing similarity metrics to the audit selection procedures and demonstrate how similarity measures can be incorporated into the selection process. The existing literature (Vasarhelyi and Halper, 1991; Kogan et al., 2014; Issa and Kogan, 2014; Li et al., 2016; No et al., 2019) does not consider the similarities among exceptions that are selected for further audit review. Without such features, audit selection models would include exceptions that are subject to the same set of risks. Reviewing multiple exceptions with the same risk profiles might be

---

\(^1\) Different distance metrics are available in the data analytics domain. The discussion of these metrics and audit implications of their use for aggregation of risk values are covered in Chapter 1.

\(^2\) There is no published literature on analyzing the possible relationships among filters. I use a payroll dataset in Chapter 1 to explain such a phenomenon in detail.
a suboptimal use of limited audit resources. I propose a “similarity check” during the selection among exceptions to keep their risk profiles diverse.

The exploration of available similarity measures in data science has motivated me to develop two selection algorithms that use the similarity of exceptions during selection. The Skipper and the Stretcher algorithms use cross similarity and risk scores during selection. The former prioritizes the risk score of exception while setting a threshold for cross similarity of selected exceptions. The latter “stretches” the cross dissimilarity of exceptions that are deemed to be risky for a given selection size. I evaluate the algorithms on synthetic and real-life datasets.

In the third chapter, I propose using feedback from confirmed instances of misstatements detected by exceptions and outliers analyses to update or generate rules for additional filter discovery. The current literature (Li et al., 2016) addresses filter rule additions through further analyses of exceptions caught by rules that represent risks the auditors are aware of. I expand this feedback loop of the continuous auditing to the instances that are exceptions and outliers and are confirmed to be misstatements. This addition to the analytics of the continuous audit (Kogan et al., 2014; Li et al., 2016) advises auditors about risks they are not aware of and recommends new filters that should be added to the transaction verification module.

Previous scholars in continuous audit (Kogan et al., 2014; Issa and Kogan, 2014; Li et al., 2016; No et al., 2019) have contributed to the literature by extending the application of advances in the data analytics domain to the continuous audit framework. This dissertation will contribute to the literature by improving the utilization of risk scores in transaction verification, allowing the users to decide how deeply they prefer to focus on
balanced or unbalanced filter values. During the selection, the use of similarity measures assures the diversity of risk presented by exceptions. Finally, analyses of confirmed instances of outliers allow the users to develop additional useful rules to catch exceptions, the rules that the users are not aware of. The proposed extensions have the potential to become an important component of the continuous audit framework.
CHAPTER 1: DISTANCES AND ANGLES OF RISK SCORE

Risk-based approaches are becoming more central to the audit domain as developments in information technologies allow more processing power and data storage and faster connection at a lower cost. Under a risk-based audit, auditors would spend more effort and resources on the accounts associated with higher risk than other accounts, leading to more effective and efficient audits (Bell et al. 2005; Rittenberg and Schwieger 2005; Knechel 2007). Similar approaches can be applied at a transaction, record or document level, increasing the quality of the audit outcome. The 2012 Audit Sampling Guide of AICPA states that items in a sampling population greater than tolerable misstatement should be removed from the population as they can present high risk and are tested separately before applying sampling. According to Vasarhelyi and Halper’s (1991) continuous audit framework, each item in the population can be evaluated from different risk perspectives or risk filters. Later extensions (Issa. 2013; No et al., 2019) of this framework use the weighted sum method to aggregate the values from risk filters to find the overall risk score for each item. These risk filters should evaluate the population of items for all risk factors that the auditor is aware of. As the items can be evaluated from a significant number of risk perspectives, during the aggregation of these risk factors into a single risk score for each item, auditors should consider two important issues: scaling and relationships among filters.

Literature Review

After starting to implement the continuous audit framework (Vasarhelyi and Halper, 1991) into different business settings, researchers reported an overwhelming number of exceptions as a result of the automatic transaction verification component of the
framework. Auditor’s inability to properly handle all exceptions may delay significantly or even make it impossible to derive timely and reliable audit evidence.

Alles et al. (2006), who refer to the problem as alarm flood, provide two possible reasons for the phenomenon as a result of their implementation of the continuous audit to certain business process controls of Siemens: sub-optimal business process control settings and overly conservative configuration of the system. In Alles et al. (2008), researchers argue that the main reasons for a significant number of alarms are the complexity of the system and its settings changes. They propose the automation of the alarm review process through the classification hierarchy. Kogan et al. (2014) are the first scholars to apply analytical continuous auditing to transaction-level systems. In their model, automatic transaction verification reports exceptions and automatic analytical monitoring reports anomalies from the data of the records. Issa and Kogan (2014) also discuss the identification of exceptions and stress the importance of prioritizing identified exceptions for audit efficiency and effectiveness. Their model offers an outlier ranking and prioritization using probabilistic ratios and differences. Brown-Liburd (2015) generalizes the issue by noting it as information overload produced by data analytic tools output. Issa (2013) also argues that although most statistical and machine learning techniques identify exceptions, they do not help much with the analysis of these exceptions. He proposes a rule-based model with a weighting system derived from experts’ knowledge as a solution for the prioritization of exceptions. On the contrary, Li et al. (2016) use belief functions to assign risk scores to exceptions.

Finally, the recently proposed model, Multidimensional Audit Data Selection (MADS) by No et al. (2019), uses several steps to derive a prioritized list of the riskiest
transactions from the whole population of transactions. In each step, the number of transactions in the set is reduced and the average risk of the set is increased. In the last step, risk scores are calculated to be used for prioritization.

Most of the above-mentioned scholars calculate the risk score using the sum of weighted filter values. However, one element that is missing in these proposed models is that relationships among filter values are not studied and do not affect risk scores. In the case of some filters assessing the same risk factors, it would be incorrect to incorporate the values of these filters to the risk score without adjustments or discounts, as it would result in an over-assessment of risk by risk scores. The model proposed in this paper would eliminate this problem by using Mahalanobis distance\(^1\) as a risk score measure.

**Risk Composition and Risk Space**

Issa (2014) and No et al. (2019) define their risk (suspicion) score of a transaction as aggregated weighted filter values, whereas Li et al. (2016) use belief functions to derive such values from transaction records. However, no previous literature elaborates on the set of filter values and how to understand or use them analytically. I regard this set of values as a *risk composition* of a record. For each record, there can be a set of filters that can be applied and a set of filter values for each record can be obtained. Although some literature (Issa 2014, Kogan 2014, Li et al., 2016 and No 2019) uses binary values (0-pass, 1-fail) as filter results, nonbinary values can also be used to estimate the risk scores more accurately and precisely. Using non-binary values for filter results captures the filter violations at different extremities, rather than treating them the same in regard to the risk factors of the

\(^1\) The Mahalanobis distance is a measure of the distance between two points in a multi-dimensional data space. It measures the distance in standard deviations and also takes the relationships between variables into account.
filters. Allowing such a wide range of values for filter results enables more data analytics methods to be applied to these values.

To understand the differences in risk compositions of transactions, consider the filter values of the transaction records given in Table 1. To simplify the case, filter values are kept between 0 (no risk) and 1 (high risk). Such values for filters can be achieved through the normalization of values.

<table>
<thead>
<tr>
<th>Transact #</th>
<th>Filter 1</th>
<th>Filter 2</th>
<th>Filter 3</th>
<th>Filter 4</th>
<th>Filter 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1002</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Example of filter values for different transactions.

Transactions 1000 and 1001 do not have anything in common in their risk composition as their non-zero filter values belong to completely different filters. While Transact 1000 has filter values from Filter 4 and Filter 5, 1001 has filter values from Filer 1, Filter 2 and Filter 3. So, these transactions represent two different sets of risk factors. When filter values of these transactions are plotted to the space as vectors, the angle between the number vectors that represent them is 90°. In data science, this similarity can be measured through Cosine Similarity, cosine of the angle between given number vectors, which is \( \cos(\alpha) = 0 \) for transactions 1000 and 1001. This metric is well known and broadly utilized in the accounting research domain to analyze similarities and differences among source documents and even financial statements. Brown and Tucker (2011) measure similarities of different years’ MD&A of the same companies using cosine similarity to conclude on year-over-year modification. Shivaani (2021) implements Cosine
and Jaccard Similarities to conduct comparative analyses on the MD&A section of Amazon and Apple. In the audit case study, Yan and Moffitt (2019) use Cosine Similarity to measure similarities among contract documents of a retailer and demonstrate how the results of such analyses can be useful in contract audits.

Alternatively, distance metrics can also be utilized to measure the dissimilarity between these transaction data points from filter values. Some clustering (k-means) and classification (k-nearest neighbor) algorithms use various distance metrics to categorize data points into different clusters or classes. However, the question of which distance metric (Manhattan, Euclidean, Mahalanobis, Minkowski, etc.) to use for the case above requires a broad discussion as this decision may greatly affect the results of analyses that are being carried on filter values of the transactions.

Both transactions 1001 and 1002 in the example above have nonzero values from the same set of filters but at different levels. In this case, the angle between the number vectors of these transaction filter values is 0°, and Cosine Similarity is 1 ($\cos 0^\circ = 1$). As these transaction filter values are proportional to each other, number vectors generated from these values overlap and, thus, are in the same direction, representing the same set of filters at the same proportions. The distance between these data points of filter values is more than 0 as these data points do not overlap due to the difference in lengths of the number vectors of the transactions. The angle-based metrics may accurately evaluate the composition of estimated risk in proportions from each used filter but cannot provide information about levels of the risk on each filter. Contrarily, when a distance-based metric

---

2 Examples of distance metrics are discussed in the next sections.
is used as a dissimilarity measure, the value of the metric becomes closer to 0 as the data points of the transactions filters values have similar values across all filters. The distance metric, however, cannot provide information about the difference in risk composition of the transactions.

**Distance for Aggregation**

Scholars (Issa. 2013; No et al., 2019) have been using a weighted sum of filters values to calculate the risk scores for records. In this paper, I approach this task from the visual analytics point. In my framework, each filter represents a separate axis of the “risk’ space, the origin of the axes representing the risk-free position of the plot regardless of the number of filters presented in the “risk” space (Figure 1-left). Since filter values cannot be negative, only one quadrant of a regular space with nonnegative values on all axes is valid. The distance from the position of any record on this plot to the origin of the axes (denoted as O) can be used as a measure for the aggregated risk score of the record for the factors known to an auditor (Figure 1-right). The farther the position of the records from the origin, the higher risk it represents.

*Figure 1. Risk space concept (left) and distance as risk score (right).*
There can be a tolerable risk region marked around the origin of the risk space so that transactions whose filter values fall within this region can be regarded as transactions with acceptable risk levels, thus, not worthy to review in detail (as shown in Figure 2). Other transactions that have a data point from their filter values outside the tolerable risk region can be regarded as risky transactions that are worthy to review and devote audit resources.

![Figure 2. Tolerable risk region of the risk space.](image)

The phenomenon of an enormous number of exceptions reported in the literature can be depicted as an enormous number of transactions that have filter values that put them outside the tolerable risk region. An overwhelming number of such transactions usually cannot be reviewed in full detail due to the limited audit resources and time constraints. This phenomenon necessitates the utilization of ordering or ranking of transactions by their aggregated risk scores.
Although using a distance metric as a method for filter value aggregation may seem straightforward, the choice of the distance calculation requires further discussion. The data analytics literature (Tan et al., 2017) offers several distance metrics that can be applied as a tool to measure risk scores. I discuss relevant distance metrics and demonstrate how they can be used. The current literature (Issa 2014, No 2019) uses the weighted sum approach for the calculation of risk scores as shown below.

\[
\text{Suspicion Score} = \sum W_{ij}W_{ij}
\]

The risk score of a transaction in the current literature (Issa 2014, No 2019) is equivalent to the Manhattan distance (a.k.a. City Block distance) from the filter value datapoint of this transaction to the origin of the risk space. The equation on the left is a calculation of Manhattan distance between any two points in space whose features (coordinates) are used as number vectors. As was explained earlier, the risk score of an exception is the distance from its position in the risk space to the origin \(O\) with weights of the axes applied. The Manhattan distance can be rewritten as the right equation below since filter weights must be applied for each filter value, and the filter values cannot be negative.

\[
d_{CB}(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i| \quad d_{CB}(x, O) = \sum_{i=1}^{n} w_i x_i
\]

This distance measure sums up the absolute values of the number vector on each axis. As neither filter weights nor filter values can be negative and axes of the risk space represent weighted filter values, Manhattan distance would be the same as the risk score.
used in current literature (Issa 2014, No 2019). Alternatively, the Euclidean distance, which is a geometric (physical) distance between points, can also be used to measure the risk score. The first equation is a calculation of Euclidean distance between any two points, whereas the second equation is its application to the risk space with filters as axes and weights provided for filters.

\[ d_E(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \]

\[ d_E(x, O) = \sqrt{\sum_{i=1}^{n} (w_i x_i)^2} \]

As given below, both Manhattan and Euclidean distances are specific variations of the Minkowsky distance (L1, L2 and L_n^3).

\[ d_{Mink}(\vec{x}, \vec{y}) = \left( \sum_{i=1}^{n} |x_i - y_i|^r \right)^{1/r} \]

\[ d_{Mink}(x, O) = \left( \sum_{i=1}^{n} (w_i x_i)^r \right)^{1/r} \]

Any variation of Minkowsky distance requires inputs (risk filter values in the case of this dissertation) to be in the same measure unit and return results (aggregated risk score in the case of this dissertation) in a unit of the measures of these inputs. Thus, risk score calculations through the above-given methods require all risk filters to provide comparable risk measures. Although variations of Minkowsky distance are logical measures to be used

---

3 When \( n \to \infty \), this specific case of Minkowski distance is known as Supremum distance and can be written as \( L_\infty \).
as a risk aggregation method, one caveat they all have is that they do not consider the relationship among filters and do not reflect the effects of these relationships in aggregated risk scores. In cases where some filter values have high pairwise correlation among themselves, there is an underlying systematic issue resulting from a single risk factor. Calculating risk scores of transactions through a variation of Minkowsky distance may overestimate risk levels of some exceptions, as these distance metrics do not discount the results based on correlation levels among filter values.

Contrarily, the Mahalanobis distance calculates the risk score in terms of standard deviations of variables and relationships among these variables, which are filter values in our case.

\[
d_M(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}
\]

\[
d_M(\vec{x}, O) = \sqrt{(\vec{w}\vec{x})^T S^{-1} (\vec{w}\vec{x})}
\]

The conversion of individual risk scores to the unit in variance is important because filters produce risk scores with different variances, and these differences are not usually considered by auditors when weights are assigned to filters. The relationships among filters are also relevant to the calculation of the risk scores and must be handled properly. The positive relationships among filter values may overestimate the risk scores as more than one filter would be assessing the same risk factor and contributing to these scores in regard to the same risk factor. Thus, Mahalanobis distance from the position of a record on the “risk space” to the origin of this space can be used as a distance metric to aggregate the filter values for each transaction after filter weights are applied. However, it needs to be
kept in mind that, unlike Minkowsky distance, Mahalanobis distance neither requires input values in the same unit of measure nor produces results in the unit of the inputs. Rather it converts filter values to the variance units for each dimension and aggregates them into a single standard deviation measure, discounting for relationships among them.

The Mahalanobis distance utilizes the inverse of a covariance matrix ($S$ in the equation) of filters values to turn risk scores into standard deviation measures and to adjust them for the relationships among these filter values. Thus, it is important to understand how the values of the elements of the given matrix affect the element values of its inverse matrix (Figure 3).

![Figure 3. Conversion of covariance matrix into an inverse matrix.](image)

If covariances among the filter values are 0 and variances of all filter values are equal to 1, the Mahalanobis distance reduces to Euclidean distance. Filters may have a wide range of variances in the continuous audit framework, but covariances are usually expected to be close to 0 if all filters represent separate risk factors. The diagonal elements of the inverse matrix would be respective reciprocals of the variances of the filters used in the framework. If the same risk factor is represented by more than one filter, the risk score of a record that has a non-zero value for these filters would be discounted by the covariances using non-diagonal elements of the inverse matrix. A simple case of only correlated two filter values is explored in Appendix B.
As with any other distance metric, Mahalanobis distance for risk score aggregation presents certain caveats. Risk score calculations are adversely affected by the filters with nonlinear relationships, decreasing the accuracy of adjustments from the correlation matrix. Auditors must also carefully assess the presence of heteroskedasticity, asymmetric skewness and multimodality in the among filter values.

**Evaluation of Distances for Risk Scores: Case Study**

The calculation of risk score through distances was applied to a real-life payroll dataset of a non-profit US organization. The general information about the dataset is given in Appendix A. Auditing textbooks (Louwers et al., 2020; Messier, Glover and Prawitt, 2021) provide common inherent risks for the payroll cycle. A set of filters were developed from these risks that are common to payroll and applied to the dataset. To evaluate the various risk factors, the set of filters were kept as diverse as possible, using different dataset fields for each filter (Table 2).

<table>
<thead>
<tr>
<th>Filter Numbers</th>
<th>Filter Names</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter 1</td>
<td>Unauthorized Other Pay</td>
<td>( \text{Other Pay} ) = \frac{2000 \text{ if Regular } = 0, \ otherwise 10% \times \text{Regular} } {25% \times \text{Regular} }</td>
</tr>
<tr>
<td>Filter 2</td>
<td>Unauthorized Overtime Pay</td>
<td>( \text{Overtime} ) = \frac{\text{Regular}}{25% \times \text{Regular} }</td>
</tr>
<tr>
<td>Filter 3</td>
<td>Unauthorized Regular Pay</td>
<td>( \text{Regular} - \text{Salary} ) if Regular &gt; \text{Salary} , otherwise 0</td>
</tr>
<tr>
<td>Filter 4</td>
<td>Unauthorized Regular Pay Increase</td>
<td>( \text{Regular} - \text{Mode(Emp.Reg.)} ) if Regular &gt; \text{Mode(Emp.Reg.), otherwise 0}</td>
</tr>
<tr>
<td>Filter 5</td>
<td>Check Issue after Termination</td>
<td>( \frac{\text{Check Amount}}{1000} ) if Check &gt; Termination + 15, otherwise 0</td>
</tr>
<tr>
<td>Filter 6</td>
<td>Duplicate Payments</td>
<td>( = \text{COUNT(By Period, By Employee)} - 1 )</td>
</tr>
<tr>
<td>Filter 7</td>
<td>Missing Adjustment for First Period for New Employee</td>
<td>( \frac{\text{Regular}}{\text{Salary}/15} - (\text{Check} - \text{Hire}) ) if 0 &lt; \text{Check} - \text{Hire} &lt; 13, otherwise 0</td>
</tr>
</tbody>
</table>

*Table 2. Filters and their formulas to calculate filter values of transactions.*
For the purposes of this dissertation, all filters are given equal weights, since learning the effects of various filter weights is not the aim of this research. Table 3 provides descriptive statistics on unweighted filter values of payroll data records.

<table>
<thead>
<tr>
<th>Risk Filters</th>
<th>count</th>
<th>mean</th>
<th>std</th>
<th>min</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter 1</td>
<td>14023</td>
<td>0.07</td>
<td>1.21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>110.75</td>
</tr>
<tr>
<td>Filter 2</td>
<td>14023</td>
<td>0.04</td>
<td>0.49</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>45.32</td>
</tr>
<tr>
<td>Filter 3</td>
<td>14023</td>
<td>0.01</td>
<td>0.11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.54</td>
</tr>
<tr>
<td>Filter 4</td>
<td>14023</td>
<td>0.01</td>
<td>0.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.45</td>
</tr>
<tr>
<td>Filter 5</td>
<td>14023</td>
<td>0.07</td>
<td>1.81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>117.44</td>
</tr>
<tr>
<td>Filter 6</td>
<td>14023</td>
<td>0.08</td>
<td>0.29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.00</td>
</tr>
<tr>
<td>Filter 7</td>
<td>14023</td>
<td>0.00</td>
<td>0.03</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 3. The descriptive statistics on unweighted filter values of payroll data records.

The dataset is made of 14023 payroll transactions, 11108 (79.21%) of which did not violate any of the filters, thus, having values of zero for all filters. 2915 exceptions with nonzero filter values for at least one filter were extracted for further analysis. The standard deviation of filter values ranges from 0.03 to 1.81 due to some extreme filter values from the filter that identifies checks issued to an employee after the employee’s termination date (max = 117.44). Table 4 provides an excerpt from the list of exceptions with the highest risk.
Table 4. Excerpt from the list of exceptions with the highest risk.

The relationship among filter values can be explored through a correlation matrix as given in Table 5.

<table>
<thead>
<tr>
<th>Transact ID</th>
<th>Filter 1</th>
<th>Filter 2</th>
<th>Filter 3</th>
<th>Filter 4</th>
<th>Filter 5</th>
<th>Filter 6</th>
<th>Filter 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>739</td>
<td>110.75</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>69.75</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>446</td>
<td>0.00</td>
<td>45.32</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6147</td>
<td>30.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>117.44</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4570</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>77.75</td>
<td>2.00</td>
<td>0.00</td>
</tr>
<tr>
<td>10042</td>
<td>0.00</td>
<td>0.00</td>
<td>5.54</td>
<td>0.00</td>
<td>0.00</td>
<td>3.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2944</td>
<td>0.00</td>
<td>0.00</td>
<td>4.89</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5355</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>62.31</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>7763</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>60.98</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2844</td>
<td>0.00</td>
<td>0.00</td>
<td>4.47</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>11599</td>
<td>43.85</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>46.78</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>740</td>
<td>38.85</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>23.43</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 5. Correlation matrix of filter values.

Although most of the pairwise relationships among filters are close to 0, the correlation between Filter 1 (Unauthorized Other Pay) and Filter 5 (Payments after Termination) is 0.56. The transactions, totalling $293,565.32, constitute 43% of the amount of all transactions that failed Filter 1 and 40% of Filter 5. Although these filters were designed to assess separate risk factors, the business processes of how payments are
issued may introduce such complexity. This is due to the fact that most of the payments after termination in significant amounts were issued as Other Pay. This is an example of where an auditor may unintentionally include several filters in the continuous auditing model that produce related values. The various dispersion levels of filter values and sufficiently strong relationships between filters may seem to make Mahalanobis distance a good candidate over any variation of Minkowski distance for aggregation.

In order to make distance metrics comparable to each other, the filter values of the extracted exceptions were normalized to the range of [0; 1]. This is also reasonable from the framework perspective since the distribution of filter values does not factor into weight assignment to these filters. However, it needs to be mentioned that the original filter values would be now represented in variances. The descriptive statistics on normalized filter values are given in Table 6.

<table>
<thead>
<tr>
<th>Risk Filters</th>
<th>count</th>
<th>mean</th>
<th>std</th>
<th>min</th>
<th>0.25</th>
<th>0.5</th>
<th>0.75</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter 1</td>
<td>2915</td>
<td>0.0029</td>
<td>0.0238</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Filter 2</td>
<td>2915</td>
<td>0.0040</td>
<td>0.0235</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Filter 3</td>
<td>2915</td>
<td>0.0044</td>
<td>0.0439</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Filter 4</td>
<td>2915</td>
<td>0.0159</td>
<td>0.0740</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Filter 5</td>
<td>2915</td>
<td>0.0028</td>
<td>0.0337</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Filter 6</td>
<td>2915</td>
<td>0.1281</td>
<td>0.1796</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.33</td>
<td>1</td>
</tr>
<tr>
<td>Filter 7</td>
<td>2915</td>
<td>0.0059</td>
<td>0.0762</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6. Descriptive statistics on the normalized filter values.

For each exception, the Manhattan and Euclidean distances were calculated from normalized filter values. The Mahalanobis distance was calculated from original unaltered filter values, as it does not require inputs to be in comparable scales. Mahalanobis distance
would be the same whether it is calculated from original or normalized filter values, as linear normalization does not affect the standard deviations of the dataset attributes.

For comparison of distance metric results, we utilized some concepts from the order theory of discrete mathematics. For each distance metric, there was produced a separate ordered list of 2915 exceptions (identified using all seven filters) from highest to lowest by their distance value as a risk score. Thus, the order of exceptions by their riskiness (using their risk scores) was produced for each of the Manhattan, Euclidean and Mahalanobis distances. For each ordered list, a separate relation matrix was constructed. Table 7 is an excerpt from the relation matrix for the exceptions ordered using Manhattan distance.

![Table 7](image)

Table 7. An excerpt from the relation matrix for the exceptions that are ordered using Manhattan distance.

In this matrix, the overall risk scores (from distance metrics) of vertically listed exceptions are compared with horizontally listed exceptions under a chosen distance metric for risk score calculation. If an exception in the vertical list has a higher risk score than the exception in the horizontal list, the cell at their intersection is assigned the value of 1. The value of -1 is assigned for the opposite case, and 0 is assigned when these exceptions have equal risk scores. It is important to note that the relation matrix above includes the relations
of any two exceptions from each exception's perspective. Moreover, diagonal elements of the matrix hold the self-comparison of exceptions, resulting in the same outcome as equivalent classes.

**Manhattan vs Euclidean.** To compare orders of exceptions obtained using different distance metrics as risk scores, an ordered list of exceptions by their Manhattan distances was taken as a base and ordered lists of exceptions under Euclidean and Mahalanobis distances were separately compared. Table 8 summarises the exceptions relation counts by their relation values when the risk score calculation method is changed from Manhattan distance to Euclidean distance.

<table>
<thead>
<tr>
<th>Manhattan</th>
<th></th>
<th>Euclidean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1</td>
<td>3,923,735</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4,138</td>
</tr>
</tbody>
</table>

*Table 8. Exception relation counts by their relation values under Manhattan and Euclidean distances.*

Manhattan and Euclidean distances are comparable to each other as their outputs are in the same units as their inputs: only 4,138 (out of over 4 million ≈ 0.1%) exception relations change the order of certain exceptions when Euclidean distance is used as risk score instead of Manhattan distance. In these 4,138 relations, 208 exceptions changed their positions to at least one higher position in the order when Euclidean distance is used instead of Manhattan, while 1,504 exceptions moved to at least one lower position. In short, these two metrics are expected to produce a similar ordered list of exceptions for a dataset of filter values.
If two transaction \((a \text{ and } b)\) switch their order relation when Euclidean is used instead of Manhattan to calculate the risk score (from \(n\) filter values), mathematically, the following can be obtained\(^4\):

\[
\begin{align*}
\left\{ \begin{align*}
\sum_n f_n^a & > \sum_n f_n^a \\
\sqrt{\sum_n (f_n^a)^2} & > \sqrt{\sum_n (f_n^b)^2}
\end{align*} \right. & \quad \longrightarrow \quad \left\{ \begin{align*}
\mu_n^a & > \mu_n^b \\
\sum_n (f_n^b)^2 & > \sum_n (f_n^a)^2
\end{align*} \right. \\
\end{align*}
\]

\[
\left. \right\} \quad \longrightarrow \quad \left\{ \begin{align*}
\frac{(\mu_n^a)^2}{\sum_n (f_n^b)^2} & > \frac{(\mu_n^b)^2}{\sum_n (f_n^a)^2} \\
\frac{\sum_n (f_n^b)^2}{n} & > \frac{\sum_n (f_n^a)^2}{n}
\end{align*} \right. \\
\end{align*}
\]

\[
\left. \right\} \quad \longrightarrow \quad \left\{ \begin{align*}
\frac{\sum_n (f_n^b)^2}{n} - (\mu_n^b)^2 & > \frac{\sum_n (f_n^a)^2}{n} - (\mu_n^a)^2 \\
\end{align*} \right. \\
\end{align*}
\]

\[
\left. \right\} \quad \longrightarrow \quad \text{var}(f^b) > \text{var}(f^a)
\]

The system of inequalities at the top simplifies into the differences in the variances of filters values for each exception. To avoid confusion with the variance of a certain filter, I refer to this variance as a horizontal variance. It shows that an exception with a lower Manhattan-based risk score than its pair can still have a higher Euclidean-based risk score, as long as it has a higher horizontal variance of filter values. This criterion is also evident

\(^4\) The first inequality refers to exceptions’ relation under Manhattan distance-based risk score, whereas the second inequality represents the relation for the same pair of exceptions that is switched under Euclidian distance.
in the horizontal variance differences of the exception pairs that switch order relation when Euclidean distance is used instead of Manhattan in one direction (Figure 4).

Figure 4. Horizontal variance differences of the exception pairs of relations that change under Euclidean and Mahalanobis distances.

The same relationship can be proved for any two variations of Minkowski distance with different orders: $L_k$ and $L_n$, where $k < n$. In fact, the effect on order relation change expands broader as the difference $n-k$ gets larger. This finding shows that the higher is the order of Minkowski distance used for risk score calculations, the more magnified is the effect of prioritizing the exceptions with risk scores that are from numerous filters rather than few. On the other hand, the lower order of Minkowski distance can be used to limit the effects of the extreme filter values on the risk score. Depending on whether an auditor wants to prioritize the review of exceptions with risk scores that are contributed by a higher or lower number of filters, auditors can decide which variation of Minkowski distance to use in risk score aggregation.
Another important decision that auditors need to make during the audit selection is the size of the selection. Although the set of the selected items may differ significantly depending on the methodology of the risk score calculations, the selection size may also contribute to this difference. I ran the selection solely based on the risk score of items for all selection sizes between 1 and 500, using Manhattan and Euclidean-based risk scores.

For each selection size, I calculated the ratio of the intersection of two selection sets. Figure 5 shows how this ratio changes with changes in selection size.

![Figure 5. The intersections portions for different selections sizes under Manhattan and Euclidean-based risk scores](image)

It can be observed that Manhattan and Euclidean based risk score selections share different portions of the elements for a given selection size. The ratio of the number of the common elements between two distance-based risk score selection to the selection size fluctuates (between 60% to 95%) for the selection sizes up to around 140, and then stabilizes (above 85%) still around 350. Auditors may receive significantly different
selection elements from using these distance metrics as a risk score for the selection sizes up to 140, but similar sets of exceptions are obtained for the selection sizes between 140 and 350. This graph may inform the auditor whether it is plausible to change the order of the Minkowski distance as a metric of the risk score for a given selection size.

*Manhattan vs Mahalanobis.* Of all variations of the Minkowski distance, the calculation of the Euclidean distance is the closest to that of the Mahalanobis distance with two main adjustments: conversion of inputs into units in variances and factoring in the relationship among filters through the inverse of filters’ covariance matrix (as provided in their respective formulas). Thus, when compared with Manhattan distance, the discussions from the previous section about horizontal variances of transactions on filter values are relevant. As this part was already covered in the previous section, I compared the order relation table from Mahalanobis distance to Euclidean distance in this section.

<table>
<thead>
<tr>
<th></th>
<th>Mahalanobis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>Manhattan</td>
<td>3,651,146</td>
</tr>
<tr>
<td>-1</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>276,727</td>
</tr>
</tbody>
</table>

*Table 9. Exception relation counts by their relation values under Euclidean and Mahalanobis distances.*

More drastic changes are observed if Mahalanobis distance is used as a risk score instead of Euclidean distance (Table 9), introducing 276,727 (≈ 6.51%) changes into the relations table. Three transactions had a maximum value Filter 6 and, thus, were assigned a value of 1 to each of these transactions during normalization of [0; 1]. Additionally, another transaction had a maximum value from Filter 4 and was also assigned 1 during
normalization. Consequently, the relations between any of the former three transactions and the latter single transaction were treated as an equivalent class for Manhattan distance-based risk score because these transactions received risk scores from a single filter. However, Mahalanobis distance-based risk score differentiates them and categorizes them to separate classes. As a result, three relations are under Manhattan distance, but not under Mahalanobis distance (Table 10).

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Filter 1</th>
<th>Filter 2</th>
<th>Filter 3</th>
<th>Filter 4</th>
<th>Filter 5</th>
<th>Filter 6</th>
<th>Filter 7</th>
<th>Normalized Manhattan Distance</th>
<th>Mahalanobis Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>3688</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.45</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>29.16</td>
</tr>
<tr>
<td>10039</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>10.37</td>
</tr>
<tr>
<td>10040</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10.37</td>
</tr>
<tr>
<td>10041</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10.37</td>
</tr>
</tbody>
</table>

Table 10. List of Exceptions that are equivalent in their risk scores under Manhattan and Euclidian but not under Mahalanobis.

To determine which factors affect the relation table, I decomposed the Mahalanobis distance into two components: the variance Effect and the covariance effect. I refer to the variance effect as changes in Mahalanobis distance due to the vertical variance of filters being other than 1. Similarly, the covariance effect is defined as changes in Mahalanobis distance due to covariance of filters being other than 0. Considering the complex inter-effect of vertical variances and covariances of filters during inverting the covariance matrix, the decomposition of the Mahalanobis distance for each transaction was performed using the procedures described in Appendix C. This decomposition showed that, out of 276,727 exception relations, 39.81% (110,154 relations) switched to opposite due to the differences in the vertical variances of filters that exceptions of relation have a value on. The differences in covariances of filters account for 23.08% (63,876 relations) of the changes in relations, while the rest of the changes are due to the combined effect of vertical...
variances and covariances. It suggests that both differences in distributions and relationships among filter values have significant effects on risk score if Mahalanobis distance is utilized for aggregation of filter values.

Comparing the lists of top 200 (≈6.86%) exceptions from the Euclidean and Mahalanobis distance-based risk score (Figure 6) showed that three-fourths (151 exceptions) of both lists were made of the same set of exceptions. However, 46 exceptions in the top 200 Mahalanobis ordered list had higher than 1000s orders under the Euclidean-based ordering, mainly due to vertical variances of the filters. The ratio of common elements of the selections based on Euclidean and Mahalanobis distances also shows that these metrics behave significantly differently for most selection sizes. The ratio fluctuates up to a selection size of 100, then steadily increases as selection size increases to 230. This auditor must carefully assess the advantages and caveats of these distance metrics if the decision comes to choose between them as a risk score aggregation method.

![Figure 6. The intersections portions for different selections sizes under Euclidean and Mahalanobis-based risk scores](image)
Summary

In this chapter of my dissertation, I introduced a risk space concept to the literature, which uses each risk metric as a separate axis to measure the transaction level items. I also explored the available distance metrics as an option to use as a risk score. As the Manhattan and Euclidean distances can be generalized into Minkowski distance, the choice of value for the order of the Minkowski distance is a key input for differences in the results. Auditors may choose the Minkowski distance with a lower order to prioritize the review of the items with risk scores from several filters over the items with similar risk scores from one or a few filters. Contrarily, a higher order of a Minkowski distance would prioritize the review of items with risk scores from one or a few filters. The use of Mahalanobis distance may help auditors to adjust the risk scores for the relationships between filters. The application of a distance metric as a risk score to the payroll dataset showed that Euclidean vs Mahalanobis can provide more significant differences than that of Manhattan vs Euclidean. The effect of the variance of filter values was substantially higher than the covariances amongst filters.
CHAPTER 2: SKIPPER AND STRETCHER ALGORITHMS FOR AUDIT SELECTION

The selection of exceptions and anomalies for audit review became a popular topic for research scholars in audit since the developments in processing power and storage of information technologies allow for the application of any number of audit tests to the whole population of records. The initial framework was proposed by Vasarhelyi and Halper (1991) as continuous process auditing, under which data is analyzed through a set of auditor-defined rules to identify exceptions or anomalies. Appelbaum, Kogan and Vasarhelyi (2017) argue that the Big Data environment requires audits to be progressive using an exception methodology.

The fact that applications of continuous auditing into practice generate an overwhelming number of exceptions and anomalies is well documented in the literature (Alles et al. 2006; Alles, Kogan, and Vasarhelyi 2008; Brown-Liburd, Issa, and Lombardi 2015, etc.). Additionally, several papers (Issa 2013; Issa and Kogan 2014; Kogan et al. 2014; Li, Chan, and Kogan 2016; No et al., 2019) propose ranking and prioritization methods of exceptions and anomalies for audit review. However, all of the available literature does not consider the similarity of the records during prioritization and ranking. This omission in the framework may result in several very similar records with sufficiently high-risk scores being selected for further review, while reviewing only one of these records may resolve the issue behind the high-risk score. Since the budget of any audit engagement allows review of a limited number of records, selecting a pertinent set of records for review is critical to the efficiency and eventually to the effectiveness of the audit. In this chapter, I propose incorporating the similarity measures, in addition to
existing risk measures, during the selection procedure. I develop two selection algorithms as a result of this approach: the Skipper and the Stretcher. These algorithms are built on previous literature on risk scores and the prioritization of exceptions. The effectiveness and robustness of algorithms were evaluated through their application on synthetic (varying in shape and/or density) and real-life datasets.

**Motivation for Research**

Upon implementing a continuous audit framework (Vasarhelyi and Halper, 1991) into different business settings, researchers reported an overwhelming number of exceptions as a result of the automatic transaction verification component of the framework (Alles et al., 2006; Alles et al., 2008; Issa., 2013; Kogan et al., 2014; Li et al., 2016). An auditor’s inability to properly handle all exceptions may delay significantly or even make it impossible to derive timely and reliable audit evidence. Most scholars (Issa., 2013; Kogan et al., 2014; Li et al., 2016; No et al., 2019) use sorted risk scores exceptions generated through a set of risk assessment functions from the records of these exceptions for prioritization and ranking. The exceptions with higher risk scores are selected to further audit review. Another methodology that was utilized by scholars (Durtschi, Hillison, and Pacini., 2004; Nigrini and Miller., 2009; Gomes da Silva and Carreira, 2013) to identify anomalous records is through an application of Benford’s law to the records amounts.

Most of the above-mentioned scholars complement the risk score with a methodology of sorting the exceptions by a degree of risk and selecting a set of exceptions with the highest scores. However, one element that is missing in these proposed models is the cross-comparison of selected exceptions for review. The number of exceptions that are selected for further audit review usually depends on audit resources and the budget
available to review these selected exceptions. Thus, it would be a misuse of audit resources to select an improper set of exceptions.

Assuming audit filters are carefully implemented, the possibility of an improper set of exceptions being selected exists even if exceptions with the highest risk scores are selected, just as it was performed in most of the previous audit selection literature (Issa., 2013; Kogan et al., 2014; Li et al., 2016; No et al., 2019). I analyzed the prioritized list of exception from a payroll data of a non-profit organization. I developed seven filters and weighted them evenly for the demonstration of the issue. Figure 7 is an excerpt from the list of exceptions sorted by Mahalanobis distance\(^1\) from the origin (used as risk score) and shows that there are several subsets of exceptions that have very similar or even identical risk compositions.

![Figure 7. Risk Composition Similarity among Exceptions Sorted by Risk Score.](image)

\(^1\) The Mahalanobis distance is a measure of the distance between two points in a multi-dimensional data space. It measures the distance in standard deviations and also takes the relationships between variables into account.
The current selection algorithms in the literature do not take cross similarity of selected exceptions into account. As a result, there is high probability in any audit setting that exceptions representing the same risk compositions are selected for further audit review. Selecting all of these exceptions for audit review might result in some audit efforts being wasted as reviews would be conducted on the same risk factors repeatedly. The sampling methods such as stratified sampling or risk based sampling focuses on only one of similarity or risk score, while ignoring the significance of the other. A solution to this problem requires a comparison of transactions while being selected and minimization of the number of similar transactions in the selection set. However, the decisions of which similarity measure to use and how to apply them to the records require further discussion. In this chapter of my dissertation, I show how similarity measures can be incorporated into the selection algorithms, so that the above described issue is avoided.

**Use of Similarity in Selection**

The data analytics domain offers several similarity measures. While angle-based similarity measures increase as the alikeness between compared items increases, the distance-based similarity measures are inversely related to the actual likeness levels between data objects. The coefficients such as Simple Matching Coefficient and Jaccard Coefficient are inapplicable due to their construction to work with binary data (Tan et al., 2017).

The purpose of the similarity measure integration into audit selection models is to avoid the over-selection of a class of records that carry similar audit risk composition.

---

2 Such as distances (Euclidean, Minkowski, Mahalanobis), angles (Cosine Similarity) and coefficients (Simple Match, Jaccard, Tanimoto).
Assuming all filter values are obtained from processing data values of the records through filter functions, similarity measures can be applied to the values of data features that are used in filter functions. As filter values of the records are based on data values, records with similar data values would also result in similar filter values. In Figure 8 below, data points of the records are plotted to the scatter plot of two of the relevant data features (fields).

Figure 8. Scatter Plot of Data Points of Records by Relevant Data Fields.

The cluster of the data points would indicate the usual values for these data fields, while outliers in the graph would indicate the position of the records with anomalous values. The random-based sampling methods tend to select mostly from the clustered records as these records constitute the majority of the population. To avoid such an issue, a requirement regarding the maximum similarity among the selected items can be implemented. Such a requirement would push the selection of some items from the clustered records to the outliers that are in the boundaries.
Data points of the record values do not represent any risk levels or indicate any risk score. The only risk information that can be derived is that, in general, outliers in the plot tend to have more audit risk than the records positioned in clusters due to their relative anomalous data values. Since no axis of the plot directly indicates risk measure, the appropriate similarity measure to apply to these data points is distance-based similarity measures. A minimum distance requirement (as maximum similarity threshold) among selected items would ensure that no records with similar risk compositions are selected. In the plot of normalized data values, Euclidean distance$^3$ is easy to use and simple to interpret.

Alternatively, the risk composition of records can be plotted in the data space whose origin represents the risk-free point and the axes represent the filters that were used to collect information about the riskiness of records. In such plot, the farther the data point is from the origin, the riskier it is in terms of assessed filters. Moreover, risk composition, which is made of filter values for each record, would indicate the contribution of each filter to the riskiness of the records. The importance of risk composition is that two records may have the same aggregated risk levels but different risk compositions. Figure 9 demonstrates the “risk space” with data points representing the risk positions of the records by two filters (Risk Filter A and Risk Filter B).

---

$^3$ Euclidean distance is a geometric distance between two data points, calculates as square root of sum of squared differences between points on each axis.
Figure 9. Scatter Plot of Record Risk Filter Values in Risk Space.

In such risk space, the similarity between any two data objects can be assessed through the angle whose sides go through these data points and the vertex is the origin of the space. The wider the angle between data points is, the lower is the similarity between these data objects. The most common angle-based similarity measure, the cosine similarity\(^4\), is inversely related to the angle between the data objects and can range from 0 (completely dissimilar) to 1 (identical). While the angle-based similarity can accurately evaluate the difference between risk compositions of the records, the major disadvantage is that it cannot differentiate risk levels.

Calculating the similarities among records is only the first step in the audit selection of exceptions. The next main step is developing an algorithm that incorporates the similarity measure and that uses it in balance with risk scores. The parameters of the algorithm are risk score (\textit{Risk\_Score}), similarity among selected exceptions (\textit{Similarity})

\(^4\) Cosine similarity is a similarity measure of two non-zero vectors, calculated as the cosine of the angle between these vectors. The normalization of the vectors does not affect their cosine similarity.
and the number of selections to be selected \((n)\). As discussed earlier, the size of the selection set for review is either decided by overall risk assessment (external audit) or constrained by limited audit resources available for use (internal audit). Thus, users have little, if any, power to choose the size of the selection, making \(n\) a constant in this setting. The audit objective is to select a subset of exceptions \((S)\) from the available set of exceptions \((C)\) identified by rule-based filters in such a way that the selected exceptions represent both diverse and significant audit risks. The objective of this model is to select the riskiest items from the candidate set, while also ensuring the dissimilarity level among selected items.

The main variables of the objective are Risk\(_{Score}\) and Similarity. In objective function, one of these variables can be optimized, while holding the other to a predefined constraint. Two respective constraints on exceptions selection can be identified as minimum individual risk level \((\text{Min}_\text{Risk}_{\text{Score}})\) and maximum cross similarity \((\text{Max}_\text{Similarity})\) of selected exceptions.

Holding either \(\text{Min}_\text{Risk}_{\text{Score}}\) or \(\text{Max}_\text{Similarity}\) constant and optimizing the other one can be reduced to a single constraint optimization problem (which is dual to the other one). User can set \(\text{Max}_\text{Similarity}\) as a constant and aim to extract \(S\) with the highest overall Risk\(_{Score}\), maximizing the riskiness of extracted exceptions (Risk Maximization) but also keeping cross similarity among these items below the similarity threshold. Alternatively, setting \(\text{Min}_\text{Risk}_{\text{Score}}\) as a constant and minimizing the Similarity among exceptions in \(S\) allows to extract exceptions with the most diverse risk (Similarity Minimization), but also ensures that all exceptions in \(S\) are above a certain risk level. Using the designations defined above, the problem of dual criteria optimization can be formalized as follows for both cases:
**Objective**  

<table>
<thead>
<tr>
<th>Risk Maximization</th>
<th>Similarity Minimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \max \sum_{i \in S} \text{Risk}_\text{Score}(e_i) )</td>
<td>( \min \sum_{(l,k) \mid l,k \in S} \text{Similarity}(e_l, e_k) )</td>
</tr>
</tbody>
</table>

**Subject to**  

| \( \text{Similarity}(e_l, e_k) \leq \text{Max}_\text{Similarity} \) | \( \text{Risk}_\text{Score}(e_i) > \text{Min}_\text{Risk}_\text{Score} \) |

**Where**  

- \( e_i, e_l \) and \( e_k \) are any exceptions selected into \( \text{Selection}_\text{Set} \) from \( \text{Candidate}_\text{Set} \).
- \( \text{Risk}_\text{Score}(e) \) is the risk score of exception \( e \).
- \( \text{Similarity}(a,b) \) is the similarity between exceptions \( a \) and \( b \).
- \( \text{Max}_\text{Similarity} \) is the similarity threshold for any pair of exceptions selected into \( S \).
- \( \text{Min}_\text{Risk}_\text{Score} \) is the risk threshold for any exception in \( S \).

*Table 11. Formalization of the similarity and risk score based optimization*

The information about data fields that is available to the user may clarify which perspective of the above described duality to use as an approach. A separate selection algorithm can be developed for each of the two approaches.

For each of the two approaches, the ideal solution would require performing aggregation or comparison of \( \text{Risk}_\text{Score} \) and \( \text{Similarity} \) values of all possible combinations of exceptions for the \( S \) with the size of \( n \) from \( C \). However, in the business world with an enormous number of transactions and records, this approach to the solution is most often computationally intractable even after the implementation of thresholds. An alternative approach is developing a greedy-heuristic algorithm, under which exceptions in \( C \) are evaluated and added to the \( S \) to achieve the best possible improvement of the objective function at every iteration. I develop a separate greedy-heuristic selection
algorithm for each approach: the Skipper for risk maximization and the Stretcher for similarity minimization.

The Skipper

The Skipper algorithm prioritizes the risk score over dissimilarity during the selection procedure while ensuring that the maximum similarity threshold is met by each of the selected exceptions. The cross similarity between any two selected exceptions should be lower than or equal to the maximum similarity threshold set by a user. With this requirement, the Skipper algorithm is expected to maximize the overall risk score of selected exceptions by as much as the similarity threshold allows. Figure 10 visually describes the overall selection procedure in the list of exceptions sorted by risk score.

Figure 10. Visualization of the Skipper Execution on a Sorted List of Exceptions.

For a given list of exceptions sorted by their risk scores, the selection starts with selecting the exceptions with the highest risk scores but skips the ones that are similar to at least one of the records that are already selected by the procedure in previous iterations.

To implement the Skipper algorithm, a user provides $Max\_Similarity$ threshold for $S$ as a constraint. The algorithm aims to extract the riskiest exceptions from a candidate set of exceptions into a selection set, in which the similarity threshold is met by all member
exceptions. First, the Skipper algorithm sorts all exceptions in $C$ by their $Risk\_Score$ from the highest to the lowest, pushing risky exceptions to the top of the list and regular - less risky ones to the bottom of the list$^5$. Then it starts the iterations of the selection procedure from the top of the sorted list of exceptions. In each iteration, a candidate exception ($e_i \in C$) from the top of the list is evaluated by its $Similarity$ to the most similar exception ($e_k \in S$) that is already included in $S$. In the first iteration, the exception at the top of the sorted $C$ is automatically included in $S$ since $S$ is empty. In the later iterations, $e_i$ is added to $S$ only if the similarity of the exception candidate $e_i$ to its most similar pair ($e_k$) in $S$ is less or equal to the $Max\_Similarity$. The candidate is removed from the $C$ as the last step of the iteration. These iterations start from the top and go down through $C$ sorted by $Risk\_Score$, each iteration evaluating the similarity of candidate exception ($e_i$) from $C$ to its most similar pair exception ($e_k$) in $S$. The selection procedure terminates when either $S$ reaches its required size ($n$) or the algorithm exhausts the $C$.

For any $C$, $S = \emptyset$, $n$ and $Max\_Similarity$, the pseudocode of the Skipper algorithm is given below:

1. Sort all exceptions in $C$ (all $e_i \in C$) by their $Risk\_Score$.
2. repeat
3. From $C$, select $e_i$ with the highest $Risk\_Score$.
4. Calculate its $Similarity(e_i, e_k)$ to each of $e_k$ ($e_k \in S$) that is already in the $S$.
5. Add $e_i$ to the $S$, if $Max(Similarity(e_i, e_k)) \leq Max\_Similarity$.
6. Remove $e_i$ from $C$.
7. until $size(S) \geq n$

It is important for the user to find the appropriate value for $Max\_Similarity$. Setting $Max\_Similarity$ too high makes the algorithm over-tolerant on the cross-similarities of

$^5$ The Skipper algorithm would run successfully even without initial sorting of exceptions by their risk scores. However, we believe this sorting helps to understand the logic of the algorithm.
selected exceptions and may result in selection with very similar exceptions, reaching its size too early without going down to the uniquely different exceptions with lower risk scores. Reviewing the exceptions with unique data features might be crucial for the audit as these records represent a different composition of risk of misstatement. Failure to review these risky records may cause material misstatements and important internal control weaknesses to go undetected. On the other hand, setting $Max\_Similarity$ too low would make the algorithm over-strict on the similarities of selected exceptions and may result in the selection procedure skipping through the important exceptions due to their similarity to the exceptions selected in previous iterations. This may even result in the intended selection size not being reached or $S$ being filled with exceptions that have very low risk scores that are not worthy to devote audit resources to. Thus, the user’s familiarity with the similarity measure and overall cross-similarity values of the exceptions may become a deciding factor in the performance of the Skipper selection algorithm. Users may have to run the selection algorithm several times with different maximum similarity thresholds or be familiar with insights and peculiarities of the data fields and values (including their cross-similarities) to be successful with a single run.

**The Stretcher**

Unlike the Skipper, the Stretcher algorithm prioritizes the dissimilarity across the selected exceptions over their risk scores during the selection procedure but also makes sure that the risk score of each selected exception is higher than the minimum risk score threshold set by a user. Considering this requirement on risk scores, the algorithm aims to minimize the similarity among selected exceptions by as much as the diversity of exceptions allows. Figure 11 describes the overall approach of the Stretcher algorithm.
First, all exceptions with Risk_Score over Min_Risk_Score threshold provided by the user are extracted from the set of all records and are saved as C. The Stretcher algorithm does not require exceptions in C to be sorted by their Risk_Score. Instead, from the unsorted C, the algorithm should select S with the required size (n) and the minimized overall cross-similarity.

Once C is defined, i.e. all exceptions with risk score over Min_Risk_Score are extracted into C, a heuristic algorithm runs in iterations, selecting the exception in C that is the most dissimilar to the exceptions in each iteration. Although the sorting of C is not required by the Stretcher algorithm, Figure 12 visually describes how the algorithm would perform on the C sorted by risk scores.

Figure 11. Steps of the Stretcher Algorithm

Figure 12. Visualization of the Stretcher Execution on a Sorted List of Exceptions.
The above-explained greedy-heuristic solution to this task implements a maximin strategy and runs in iterations. Maximin and Minimax are decision rule algorithms in computational mathematics and artificial intelligence aimed at achieving optimal point for the parameter that is affected by several variables. Dealing with gains in game theory, these algorithms can be used to find the best scenario of the worst case or the worst scenario of the best case. In the above-described audit setting, it can be utilized to select the optimal set of audit items that results in the best use of audit resources. As discussed above, two relevant variables for audit selection are risk score and similarity to the already selected items. In the iterations of the Stretcher algorithm, each exception is evaluated through its similarity to its most similar pair that is already in the selection set. This procedure can be accomplished through running minimax (select the item with min similarity to its counterpart with max similarity) selection strategy.

As the Min_Risk_Score threshold is applied before the selection iterations of the Stretcher algorithm start to run, the risk scores of exceptions in C do not play any role during the selection procedure. In each iteration, all exceptions in C would be evaluated only by their similarity to their most similar counterpart in S and the exception with the least similarity to their most similar counterpart in S is also selected into S.

After all exceptions with Risk_Score over Min_Risk_Score are extracted from the population of records, thus creating C, two exceptions with the lowest pairwise Similarity are selected and added to S. Then, in each iteration, each of the exceptions (e_i) in C is evaluated through their Similarity to their most similar counterpart (e_k) in S. The e_i with the

---

6 Minimax was used by Nicolaus Bernoulli in zero-sum two-person game in his response letter to Pierre Remond de Montmort’s. Montmort’s “Essay d’analyse sur les jeux de hazard” published in 1713 contains correspondence on probability problems between Montmort and Bernoulli.
lowest \textit{Similarity} to its most similar counterpart in \( S \) is added to \( S \) as the next exception.
The selection procedure terminates when \( S \) reaches its required size \( (n) \).

For any \( C = \emptyset, S = \emptyset, n \) and \textit{Min\_Risk\_Score}, the Stretcher algorithm is given below:

1: Extract all exceptions \( (e_i) \) with \( \text{Risk\_Score} \geq \text{Min\_Risk\_Score} \) and save them into \( C \).
2: Select the most dissimilar pair of \( e_i \) from \( C \) and add them to \( S \). Remove both \( e_i \) from \( C \).
3: \textbf{repeat}
   4: For each \( e_i \) in \( C \) assign its counterpart \( e_k \) in \( S \) where the \textit{Similarity} value is maximized.
   5: Select \( e_i \) from \( C \) with minimum assigned \textit{Similarity} in the previous step.
   6: Add \( e_i \) to \( S \) and remove \( e_i \) from \( C \).
7: \textbf{until} \( \text{size}(S) \geq n \)

As was required with the Skipper, to apply the Stretcher algorithm appropriately, users should be familiar with the overall risk score levels of the records in the population dataset and the \textit{Min\_Risk\_Score} for a record to have audit resources devoted for review. Setting the \textit{Min\_Risk\_Score} too high may leave some risky records out of \( C \) and leave them unreviewed. Furthermore, if the size of \( C \) is too small and includes exceptions with high similarity, the Stretcher algorithm may select very similar exceptions into \( S \). Contrarily, setting the \textit{Min\_Risk\_Score} too low would make the algorithm include some exceptions that are unworthy to be reviewed into \( C \) and eventually include them into \( S \), wasting audit resources with unnecessary reviews. The knowledge about the optimal risk score levels of records might be gained through analyses of the data and/or an application of the selection algorithms several times. The information about the distribution of exceptions by aggregated risk score and/or by filter values may help to gain such knowledge. Additionally, the users should keep in mind that risk score levels and cross-similarities of exceptions are not fixed but may change throughout the periods. This requires the
thresholds for maximum similarity and minimum risk score of exceptions to be reviewed periodically and adjusted.

**Evaluation of Algorithms: Application on Synthetic Data**

To evaluate the behavior of the algorithms described above, two different validation approaches were utilized. Under the first experimental approach, four synthetic datasets differing in either shape or density were created to evaluate the robustness and invariance of the selection algorithms to the changes in density and shape of the datasets. The points of all datasets were chosen from the first quadrant of the two-dimensional plane for visualization purposes. The first “square” dataset was created by using as a datapoint each integer point of the 100 by 100 square, thus creating 10201 data points. The Rightward dataset was also created in a similar manner by evenly removing some points from the top left half of the dataset. The round dataset was created from the Square dataset by removing all points outside of the circle that has a center at (50, 50) and a radius of 50. The Square-Round dataset was created from the Square dataset by evenly removing half of the points outside of the circle that has a center at (50, 50) and a radius of 40. Each point was assigned a risk score equal to its distance from the origin of the plane. The scatter plot of the datasets (Figure 7) and the descriptive statistics (Table 12) are given below.

![Scatter Plot of Synthetic Datasets](image)

*Figure 13. The Scatter Plot of Synthetic Datasets Used for Evaluation.*
I decided to use Euclidean Distance as a dissimilarity measure in the evaluation section because of its applicability. To evaluate the dissimilarity of transactions during the selection procedure, data fields that were key inputs to the filters to identify exceptions were normalized. The distances between each pair of exceptions were calculated and were saved in a form of a dissimilarity matrix. Figure 14 presents how an exception selection without considering similarity would perform on these synthetic datasets. As it can be observed, the selections represent the available points of the datasets from the upper right corners, as these points carry higher risk scores than points from other parts of the datasets.

Table 12. Descriptive Statistics of Datapoints of the Synthetic Datasets.

<table>
<thead>
<tr>
<th></th>
<th>Even</th>
<th>Rightward</th>
<th>Round</th>
<th>Square - Round</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Points</td>
<td>10201</td>
<td>7376</td>
<td>7845</td>
<td>7613</td>
</tr>
<tr>
<td>Evenly Distributed</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Shape</td>
<td>Square</td>
<td>Square</td>
<td>Round</td>
<td>Round</td>
</tr>
<tr>
<td>Maximum Distance</td>
<td>141</td>
<td>141</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Max x</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Max y</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Min x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Min y</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 14. The Scatter Plot of Selected Exceptions from the Synthetic Datasets without Use of Similarity

The Skipper. When Euclidean Distance is used as a dissimilarity measure instead of a similarity, the shorter the distance between data points, the higher the similarity
between these items. Therefore, a user provides a minimum threshold for Euclidean Distance between data points of records as a dissimilarity measure. We replaced \textit{Max\_Similarity} with \textit{Min\_Distance} in the algorithm and adjusted the pseudocode of the algorithm accordingly.

The Skipper algorithm was run on these synthetic datasets with parameters of acceptable minimum distance to the closest neighbor in the selection being \textit{Min\_Distance} = 10 and selection size being \( n = 100 \). The scatter plot of selected data points (Figure 15) and the descriptive statistics on the distance to the closest neighbor of the selected data points (Table 13) by the algorithm is given below.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{scatter_plot.png}
\caption{The Scatter Plot of Selected Exceptions from the Synthetic Datasets using the Skipper Algorithm.}
\end{figure}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
 & \textbf{Even} & \textbf{Rightward} & \textbf{Round} & \textbf{Square-Round} \\
\hline
\textbf{Selected} & 100 & 100 & 88 & 100 \\
\hline
\textbf{Mean} & 10.04 & 10.02 & 10.04 & 10.01 \\
\hline
\textbf{Standard Deviation} & 0.11 & 0.09 & 0.08 & 0.04 \\
\hline
\textbf{Minimum} & 10.00 & 10.00 & 10.00 & 10.00 \\
\hline
\textbf{25\%ile} & 10.00 & 10.00 & 10.00 & 10.00 \\
\hline
\textbf{50\%ile} & 10.00 & 10.00 & 10.00 & 10.00 \\
\hline
\textbf{75\%ile} & 10.00 & 10.00 & 10.05 & 10.00 \\
\hline
\textbf{Maximum} & 10.63 & 10.63 & 10.44 & 10.30 \\
\hline
\end{tabular}
\caption{The Descriptive Statistics on Distances to Closest Selected Neighbor of Selected Exceptions from the Synthetic Datasets using the Skipper Algorithm.}
\end{table}
The scatter plot shows that selected data points are evenly distributed throughout the synthetic datasets, leaving some empty space in the lower-left part of the three (square) datasets. This is expected since the Skipper selects items in iterations starting from the top right corner of the dataset, where the riskiest data points reside. The selection algorithm keeps the distance among selection data points at 10 as long as there is a data point available to select and terminates once the selection size reaches 100. Thus, most of the selected data points have a distance of 10 to their closest selected neighbor and we believe that distances between selected data points that are slightly over 10 are due to the unavailability of datapoints between given items in the population. This can be confirmed by the low level of the standard deviations (0.04 to 0.11) and range (0.30 to 0.63) of distances to the closest neighbor values.

In the case of the round shape dataset, only 88 data points were selected, because the \textit{Min\_Distance} = 10 parameter value would not allow adding another point to the selection as this would make the distance between some selected points less than 10. Thus, the Skipper algorithm terminates once the selection size has reached 88.

\textit{The Stretcher.} As explained earlier, if Euclidean Distance (\textit{Distance}) is used as a similarity measure, the similarity between two records decreases when the distance between data points of these records increases. Therefore, in each iteration of the Stretcher algorithm, \( e_i \) with the farthest distance to its closest pair in \( S \) should be selected and added to \( S \). This application of the distance metric requires implementing maximin (select the item with \textit{max} distance to its closest counterpart with \textit{min} distance) instead of the minimax strategy discussed in the previous section.
The Stretcher algorithm was also similarly applied with parameters minimum risk score \( \text{Min\_Risk\_Score} > 0 \) and selection size \( n = 100 \). Given below are the scatter plot of selected data points (Figure 16) and the descriptive statistics on the distance to the closest neighbor of the selected data points (Table 14) by the Stretcher algorithm.

![Figure 16. The Scatter Plot of Selected Exceptions from the Synthetic Datasets using the Stretcher Algorithm.](image)

<table>
<thead>
<tr>
<th></th>
<th>Even</th>
<th>Rightward</th>
<th>Round</th>
<th>Square-Round</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Selected</strong></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td><strong>Mean</strong></td>
<td>9.93</td>
<td>9.91</td>
<td>8.33</td>
<td>9.83</td>
</tr>
<tr>
<td><strong>Standard Deviation</strong></td>
<td>1.6</td>
<td>1.55</td>
<td>1.04</td>
<td>1.53</td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
<td>8.49</td>
<td>8.49</td>
<td>7.28</td>
<td>8.49</td>
</tr>
<tr>
<td><strong>25%ile</strong></td>
<td>8.6</td>
<td>8.6</td>
<td>7.62</td>
<td>8.60</td>
</tr>
<tr>
<td><strong>50%ile</strong></td>
<td>9.22</td>
<td>9.22</td>
<td>8.06</td>
<td>9.22</td>
</tr>
<tr>
<td><strong>75%ile</strong></td>
<td>12</td>
<td>12</td>
<td>8.54</td>
<td>12.00</td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td>13</td>
<td>13</td>
<td>11.66</td>
<td>13.04</td>
</tr>
</tbody>
</table>

*Table 14. The Descriptive Statistics on Distances to Closest Selected Neighbor of Selected Exceptions from the Synthetic Datasets using the Stretcher Algorithm.*

As the algorithm of the Stretcher is based on a greedy-heuristic approach, the selection is locally optimal but less optimal globally when compared to the Skipper algorithm. This can be observed by the higher standard deviation (1.04 to 1.6) and range (4.38 to 4.55) of distances of selected data points to their closest neighbors.
The evaluation of algorithms on synthetic data confirms both Skipper and Stretcher algorithms to be effective in preventing the data points with similar values being selected. The density and shape of the datasets could not affect the results of the selection procedure adversely. However, for successful execution of the algorithms, users still need to provide a proper threshold value (*Min_Distance* for Skipper or *Min_Risk_Score* for Stretcher) depending on the distribution of the dataset.

**Evaluation of Algorithms: Application on Real-Life Data**

The second validation approach is the case study of a payroll dataset from a nonprofit US organization through the application of both selection algorithms. The dataset information is given in Appendix A. Seven filters were developed and the risk score for all records was calculated using Mahalanobis Distance, as was covered in the previous chapter of this dissertation. The dissimilarity matrix for all records was built using Euclidean Distance from the normalized values of the dataset fields used in filters as inputs. The risk score based on Mahalanobis Distance and dissimilarity measure based on Euclidean Distance are two key inputs to the Skipper and the Stretcher algorithms.

**The Skipper.** From 14023 records of the payroll dataset, 2915 exceptions with a risk score of more than 0 were identified and were included in *Ex_Set*. As explained earlier, the Skipper algorithm requires some knowledge about exception distribution on the distance to the closest exception to set a proper level of a threshold for this feature of exceptions. Since the *Ex_Set* includes all exceptions that have a risk score other than 0, it is expected that exceptions with very low risk are also in this list. These “unworthy to review” exceptions may even constitute a majority of all exceptions. Thus, for the Skipper algorithm to be effective, two objectives must be achieved: (a) the minimum risk score in
the Sel_Set must be large enough to constitute a risk, so that audit resources can be devoted, and (b) the Sel_Set must be selected after considering a large enough portion of Ex_Set as a candidate for Sel_Set, so that algorithm has a broad range of exceptions to choose from to fill the Sel_Set with diverse exceptions. The histogram (Figure 17) of exceptions by their Euclidean distance to another closest exception was used for this purpose.

Figure 17. The Histogram of Exceptions in Payroll Dataset by Euclidean Distance to the Closest Exception.

As the graph shows, the closest neighbor of most exceptions lies within 0.1 Euclidean distance in normalized data values used for filters. Thus, the Skipper algorithm was executed for the thresholds of minimum Euclidean distance to the closest selected exception with different values between 0 and 0.25.

Table 15 provides the results of the algorithm for each run. The run of the algorithms for the dissimilarity threshold of 0 represents the selection of exceptions without consideration of similarity among selected exceptions. The comparison of the results from
other runs to the first column shows how efficiently the Skipper algorithm selected dissimilar and risky exceptions as the dissimilarity threshold increases.

<table>
<thead>
<tr>
<th>Threshold for Dissimilarity</th>
<th>0</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
<th>0.16</th>
<th>0.17</th>
<th>0.20</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>80</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>% Processed</td>
<td>6.7%</td>
<td>6.7%</td>
<td>18.7%</td>
<td>41.7%</td>
<td>58.4%</td>
<td>97.8%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>% Unprocessed</td>
<td>93.3%</td>
<td>93.3%</td>
<td>81.3%</td>
<td>58.3%</td>
<td>41.6%</td>
<td>2.2%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Processed</td>
<td>100</td>
<td>196</td>
<td>544</td>
<td>1,217</td>
<td>1,702</td>
<td>2,851</td>
<td>2,915</td>
<td>2,915</td>
</tr>
<tr>
<td>Unprocessed</td>
<td>2,815</td>
<td>2,719</td>
<td>2,371</td>
<td>1,698</td>
<td>1,213</td>
<td>64</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Min SusScore</td>
<td>10.61</td>
<td>6.92</td>
<td>3.46</td>
<td>3.46</td>
<td>1.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Distance to the Closest Selected Exception**

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>0.091</th>
<th>0.121</th>
<th>0.164</th>
<th>0.223</th>
<th>0.242</th>
<th>0.246</th>
<th>0.288</th>
<th>0.363</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>std</td>
<td>0.149</td>
<td>0.127</td>
<td>0.123</td>
<td>0.148</td>
<td>0.151</td>
<td>0.149</td>
<td>0.153</td>
<td>0.170</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>0.000</td>
<td>0.051</td>
<td>0.101</td>
<td>0.150</td>
<td>0.161</td>
<td>0.170</td>
<td>0.202</td>
<td>0.251</td>
</tr>
<tr>
<td>25%</td>
<td>0.008</td>
<td>0.057</td>
<td>0.105</td>
<td>0.157</td>
<td>0.171</td>
<td>0.177</td>
<td>0.210</td>
<td>0.262</td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>0.030</td>
<td>0.068</td>
<td>0.122</td>
<td>0.171</td>
<td>0.179</td>
<td>0.187</td>
<td>0.227</td>
<td>0.294</td>
<td></td>
</tr>
<tr>
<td>75%</td>
<td>0.100</td>
<td>0.128</td>
<td>0.161</td>
<td>0.219</td>
<td>0.228</td>
<td>0.221</td>
<td>0.271</td>
<td>0.371</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>0.780</td>
<td>0.780</td>
<td>0.780</td>
<td>1.002</td>
<td>0.901</td>
<td>0.901</td>
<td>0.901</td>
<td>0.901</td>
<td></td>
</tr>
</tbody>
</table>

*Table 15. Descriptive Statistics of Selection Exceptions from Payroll Dataset Using the Skipper Algorithm with Different Dissimilarity Thresholds.*

As it can be observed from the results, the Skipper algorithm goes through and considers about 50% of the exceptions (% of Exceptions Covered) in the sorted C for Min_\_Distance between 0.15 and 0.16. It is also important to note that the minimum risk score in S becomes closer to 1 as Min_\_Distance goes over 0.16. As risk scores of exceptions are calculated using Mahalanobis Distance on filter values, it is measured in terms of standard deviation adjusted for covariances among filters. Thus, the value of the minimum risk score
in $S$ should be no less than 2. The line graph in Figure 18 shows how the minimum risk score in $S$ drops as the user increases the value for the $Min\_Distance$ parameter of the Skipper algorithm.

![Graph showing the relationship between Similarity Threshold (Min_Distance) and Minimum Risk Score of Selected Exceptions from the Payroll Dataset using the Skipper Algorithm.](image)

Figure 18. Relationship between Similarity Threshold (Min\_Distance) and Minimum Risk Score of Selected Exceptions from the Payroll Dataset using the Skipper Algorithm.

The analyses above show that the preferred input value for $Min\_Distance$ threshold to run the Skipper Algorithm on the provided payroll dataset is between 0.15 and 0.16. The $Min\_Distance$ of less than 0.15 may leave out some exceptions with a risk score higher than 3 from $S$, while the $Min\_Distance$ of higher than 0.16 results in including in $S$ some exceptions that have the risk score close to 0.

**The Stretcher.** As was mentioned earlier, 2915 records, out of 14023, had a nonzero risk score and most of these exceptions may not carry significant risks based on their low risk scores. Thus, it is important to set a proper value for $Min\_Risk\_Score$ in the Stretcher
algorithm so that the exceptions in $C$ have an adequate level of risk and enough diversity for selection. The distribution of exceptions by their risk scores (Table 16) may give the first insights for this task.

<table>
<thead>
<tr>
<th>Risk Score Range</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0 - 1)</td>
<td>199</td>
<td>41.13%</td>
</tr>
<tr>
<td>[1 - 2)</td>
<td>213</td>
<td>7.31%</td>
</tr>
<tr>
<td>[2 - 3)</td>
<td>124</td>
<td>4.25%</td>
</tr>
<tr>
<td>[3 - 4)</td>
<td>978</td>
<td>33.55%</td>
</tr>
<tr>
<td>[4 - 5)</td>
<td>93</td>
<td>3.19%</td>
</tr>
<tr>
<td>[5 - 10)</td>
<td>196</td>
<td>6.72%</td>
</tr>
<tr>
<td>[10 - 50)</td>
<td>107</td>
<td>3.67%</td>
</tr>
<tr>
<td>[50 - 100)</td>
<td>5</td>
<td>0.17%</td>
</tr>
</tbody>
</table>

*Table 16. Distribution of Exceptions in the Payroll Dataset by Risk Scores.*

As Mahalanobis Distance based risk score measures risk levels in standard deviations (adjusted for correlation among variables), it is not recommended to select the exceptions with risk scores less than 1. Most risky exceptions would have a risk score of more than 3. Taking into account the distribution of exceptions by risk score, the Stretcher algorithm was run with $Min_Risk_Score$ values between 1 and 4, as shown in Table 17. For comparison, the results of the selection without considering the similarity among exceptions is provided in the last column of the table (risk score threshold of 10.61).
The distance between closest exceptions among the selection of the Stretcher algorithm does not change much for Min_Risk_Score values between 1 and 3, but drops significantly after Min_Risk_Score ≥ 3. This is shown in the line graph of Figure 19, which represents the results of 30 runs of the Stretcher algorithm for values 1 ≤ Min_Risk_Score ≤ 4.

<table>
<thead>
<tr>
<th>Risk Score Threshold</th>
<th>0.5</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
<th>2.5</th>
<th>3</th>
<th>3.5</th>
<th>4</th>
<th>10.61</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate Exceptions</td>
<td>1880</td>
<td>1716</td>
<td>1587</td>
<td>1503</td>
<td>1434</td>
<td>1379</td>
<td>498</td>
<td>401</td>
<td>100</td>
</tr>
<tr>
<td>count</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>% of exceptions</td>
<td>64.49</td>
<td>58.87</td>
<td>54.44</td>
<td>51.56</td>
<td>49.19</td>
<td>47.31</td>
<td>17.08</td>
<td>13.76</td>
<td>3.43</td>
</tr>
<tr>
<td>Distance between closest</td>
<td>0.166</td>
<td>0.162</td>
<td>0.161</td>
<td>0.161</td>
<td>0.159</td>
<td>0.159</td>
<td>0.100</td>
<td>0.087</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Risk Score of Selected Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
</tr>
<tr>
<td>min</td>
</tr>
<tr>
<td>25%</td>
</tr>
<tr>
<td>50%</td>
</tr>
<tr>
<td>75%</td>
</tr>
<tr>
<td>max</td>
</tr>
</tbody>
</table>

Table 17. Descriptive Statistics of Selection Exceptions from Payroll Dataset Using the Stretcher Algorithm with Different Risk Score Thresholds.
Figure 19. Relationship between Risk Score Threshold (Min_Risk_Score) and Similarity (Distance) of Selected Exceptions from the Payroll Dataset using the Stretcher Algorithm.

Thus, Min_Risk_Score to run the Stretcher algorithm is recommended to be not higher than 3.4 to allow diversity but also not lower than 1 to allow higher risk. It is also important to mention that more than 40% of the exceptions in the current dataset have risk scores lower than 1 and, thus, do not constitute significant risk according to developed filters.

The evaluation analyses of both algorithms through their implementation on the payroll dataset show the importance of the knowledge about the distribution of input values that are used as thresholds. Setting the thresholds too high or too low may either exclude some risky exceptions from selection or include some not risky enough exceptions to the selection. While excluding risky exceptions affects the effectiveness of the continuous audit framework, including exception with low risk would affect its efficiency.
Modification for Special Cases

The implementation of the above-introduced algorithms in real-life cases may require some modification to the steps of the algorithm or adjustments to the logic of the algorithm. In some cases, due to external information or auditor’s own knowledge, there might be a subset of exceptions with few records that the auditor must review regardless of their risk scores or similarity among them. In these cases, the auditor may want to be confident that each of these records is included in $S$ in addition to other exceptions selected by the algorithm. Both Skipper and Stretcher algorithms can be modified for this case just by starting $S$ with the set of these exceptions (rather than starting $S$ with $\emptyset$) and removing them from the population of records before the selection algorithms start to run.

Some records may have high risk scores however the auditor may wish to not review them as s/he may already know why these records have false high-risk scores. The modification of the risk score generating model and its rerun might not be a feasible option for an auditor in such circumstances. In these cases, the auditor needs to manually remove these records from the population of records before the algorithms are executed.

Summary

In this chapter of my dissertation, I explored the current literature of audit selection models and developed two selection algorithms that use the similarity of exceptions during selection in addition to risk scores. Both the Skipper and the Stretcher algorithms use similarity and risk scores. The former prioritizes the risk score of exception while setting a threshold for cross similarity of selected exceptions. The latter “stretches” the cross dissimilarity of exceptions that are deemed risky for a given selection size. The evaluation of the algorithms on synthetic and real life datasets showed that, although these algorithms
use different approaches for selection, the auditor’s domain knowledge of the dataset used during selection is crucial for the algorithms to achieve their purpose.

Future research could attempt to add more parameters to the selection algorithms in addition to cross similarities and risk scores of exceptions. Among those parameter candidates can be the materiality of exceptions. Additionally, researchers could explore other selection strategies to develop new algorithms for the selection of exceptions that comply with similarity requirements set by the auditor.
ESSAY THREE: EXCEPTION RULE GENERATION FROM OUTLIERS

The continuous auditing framework (Vasarhelyi and 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 on business processes for the attention of responsible personnel. Later designs of transaction-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 responsible personnel. An auditor may interpret the exceptions alarms as his risks 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 any specific audit risk factor but flag the accounting items with significantly different data values than usual ones. The review of these alarms by responsible personnel may reveal the true nature of underlying accounting items, importantly, whether they are fraudulent or erroneous. Continuous audit literature discusses how the outcome of reviews can be used to improve the models by refining the parameters (Kogan et al., 2014) or including additional rules (Li, Chan, and Kogan, 2016). However, the interoperation of transaction verification and analytical monitoring is rarely discussed in the 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
risk factors. 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.

In this chapter of my dissertation, I propose adding a rule induction component to the existing framework of the continuous audit between transaction verification and outlier/anomaly detection. This component analyzes the confirmed outliers and exceptions of the provided dataset and recommends additional possible filters for the 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 that was used in previous chapters of this dissertation.

**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). Since auditors are not expected to have absolute knowledge about all audit risk factors, 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 20). 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 the transaction-level data to extract transactions with anomalous data values.

Figure 20. The architecture of the Continuous Data Level Auditing System (adapted from Kogan, Alles, Vasarhelyi, and Wu, 2014)

In another paper on the implementation of continuous audit, Li, Chan, and Kogan (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 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

---

\(^1\) Repeated Incremental Pruning to Produce Error Reduction
extensively by authors, they limit the feedback loop of the model to the cycle of exceptions 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 may rather receive more useful feedback from the positive instances of misstatements identified by the analytical monitoring component of the continuous auditing framework (Figure 21). The feedback about confirmed misstatements from anomalies and outliers may provide more knowledge about other business rules and controls missing in transaction verification.

![Figure 21. Feedback to Transaction Verification from Confirmed Misstatements of Anomalies and Outliers.](image)

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

**Risk Awareness vs. Risk Exploration.** Although not explicitly stated in most continuous audit framework applications literature, the allocation of audit resources to exploration and exploitation\(^2\) of audit 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 minimizes the loss by implementing the auditor's understanding of risk through predefined filters. On the other hand, the outlier/anomaly monitoring layer serves as an exploration module to gather additional knowledge about audit 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 audit risk factors. 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 what portion of available audit resources to dedicate for 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

---

\(^2\) The exploitation and exploration are the central modules of reinforcement learning in machine learning. Exploitation refers to learning through interaction in a random environment, while exploration refers to accumulating information to optimize the objective function (Sutton and Barto 1998).
wasting resources on investigating less risky accounting items caught by filters, meanwhile, 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 filters could catch. Thus, the auditor must identify risky items while also expanding his risk awareness in the same audit setting.

**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 have certain risk factors. Each audit filter in the transaction verification should assess the population of accounting records regarding a separate audit risk factor. 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 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 misstatements, as some procedures are more manual than analytical (ex., 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 22.
Figure 22. 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 thresholds and weights used by a filter in calculating the risk score. The filter's inability to detect an accounting item that possesses the same audit risk that is material and significant and targeted by the filter might be due to inappropriate values of thresholds used in the function of the filter. Setting the thresholds 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.

In this chapter of the dissertation, I explore 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 rules of the transaction verification, partially addressing the filter ineffectiveness.

**Rule Induction from Confirmed Outliers**

Generating rules for filter recommendations from confirmed outliers data requires only positive instances of outliers to be labelled as "Misstated". The outliers' negative instances and all inliers must be marked as "Regular" (Figure 23). The confirmed positive instances of the exceptions can also be labelled as "Misstated" and added to the positive instances of the outliers to check the accuracy of thresholds in filters that caught these exceptions. 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.
While rule induction for the following period \((t+1)\) from a single period data \((t)\) is possible, the use of multiple past periods 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. Unavailability of the past period labelled data may force the user to implement the expanding window approach for the initial stages of implementation until the intended window size is reached. Figure 24 demonstrates the moving window approach-based model with a three-period window and 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.
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 moving window size may result in model overfitting or underfitting, as was explained in previous paragraphs.

Figure 25 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 $\text{max\_depth}=3$ allows having rules with no more than three attributes, Maximum Negative Class Ratio of $\text{max\_class\_ratio}=1$ allows pruning the right branch ($\text{Incentive} \geq 14093.36$) 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 ($\text{Incentive} \leq 22230.25$ and $\text{Incentive} \geq 26159.86$) to a single rule despite the other negative instances ($\text{Incentive} \leq 26159.86$) on this node.
Figure 25. Interactions of Maximum Rule Complexity, Maximum Negative Class Ratio, and Minimum Positive Instances parameters.

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 followings 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 risk factors 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 risk factors may disappear from 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 26 visualizes the dynamic of the rule merge from a data points view 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.
Having too high a negative class ratio in $a_1 < x_1 < a_2$ & $b_1 < x_2 < b_2$ makes this subspace not to 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 merge 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.

**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 in this chapter of my dissertation. 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 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\(^3\) (Breiman et al. 1984) was used as a rule induction algorithm to generate rule sets for high-risk subspaces because of the package availability in Python and the script reliability certification. A recursive rule growing procedure with parameters of max\(_{\text{dept}} = 3\), max\(_{\text{class\_ratio}} = 1\), and min\(_{\text{positives}} = 3\) was applied.

A greedy heuristic rule growing (as described in Appendix A) using the decision tree algorithm CART was implemented to obtain a ruleset for each period of the dataset. Table 18 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,

---

\(^3\) Classification And Regression Trees
most filters present relevant rules for payroll audit, such as unauthorized overpayments on regular pay, overtime pay, vacation pay, tuition pay and other pay.

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 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.
<table>
<thead>
<tr>
<th>Filter</th>
<th>Data Fields</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
<th>Period 7</th>
<th>Period 8</th>
<th>Period 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter 1</td>
<td>Tuition≥</td>
<td>75</td>
<td>75</td>
<td>65.5</td>
<td>65.5</td>
<td>65.5</td>
<td>67.97</td>
<td>67.97</td>
<td>68.97</td>
<td></td>
</tr>
<tr>
<td>Filter 2</td>
<td>Overtime≥</td>
<td>571.38</td>
<td>584.89</td>
<td>674.225</td>
<td>674.225</td>
<td>638.905</td>
<td>638.905</td>
<td>586.59</td>
<td>690.825</td>
<td>476.36</td>
</tr>
<tr>
<td>Filter 3</td>
<td>Overtime≥</td>
<td>476.36</td>
<td>1706.94</td>
<td>Regular≤</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filter 4</td>
<td>Overtime≥</td>
<td>80.595</td>
<td>80.595</td>
<td>687.5</td>
<td>687.5</td>
<td>738.465</td>
<td>738.465</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filter 5</td>
<td>Other Pay≥</td>
<td>5934.6</td>
<td>4744.91</td>
<td>5039.83</td>
<td>4850.04</td>
<td>4850.04</td>
<td>4850.04</td>
<td>4850.04</td>
<td>4850.04</td>
<td>4850.04</td>
</tr>
<tr>
<td>Filter 6</td>
<td>Incentive≥</td>
<td>14093.4</td>
<td>14093.4</td>
<td>13225</td>
<td>13225</td>
<td>13225</td>
<td>13225</td>
<td>13225</td>
<td>13225</td>
<td>13225</td>
</tr>
<tr>
<td>Filter 7</td>
<td>Vacation≥</td>
<td>609.935</td>
<td>609.935</td>
<td>507.835</td>
<td>491.48</td>
<td>491.48</td>
<td>491.48</td>
<td>491.48</td>
<td>491.165</td>
<td></td>
</tr>
<tr>
<td>Filter 8</td>
<td>Vacation≥</td>
<td>1945.47</td>
<td>1945.47</td>
<td>7284.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filter 9</td>
<td>Overtime≥</td>
<td>325.32</td>
<td>325.32</td>
<td>328.32</td>
<td>328.32</td>
<td>328.32</td>
<td>328.32</td>
<td>328.32</td>
<td>328.32</td>
<td>328.32</td>
</tr>
<tr>
<td>Filter 10</td>
<td>Other Pay≥</td>
<td>178.775</td>
<td>178.775</td>
<td>178.775</td>
<td>178.775</td>
<td>280.15</td>
<td>167.775</td>
<td>228.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filter 11</td>
<td>Regular≥</td>
<td>324.245</td>
<td>324.245</td>
<td>100.54</td>
<td>167.5</td>
<td>167.5</td>
<td>413.03</td>
<td>154.44</td>
<td></td>
<td>50000</td>
</tr>
</tbody>
</table>
The thresholds of most filters fluctuate (Figure 27), 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 27. Changes in filter threshold over time.](image)

To evaluate the 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 19). 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 the 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.
Table 19. Accuracy of filters on predicting error of following period.

<table>
<thead>
<tr>
<th>Caught (Predicted)</th>
<th>P 2</th>
<th>P 3</th>
<th>P 4</th>
<th>P 5</th>
<th>P 6</th>
<th>P 7</th>
<th>P 8</th>
<th>P 9</th>
<th>Errors</th>
<th>Exceptions</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter 1</td>
<td>3 (3)</td>
<td>4 (4)</td>
<td>4 (4)</td>
<td>9 (10)</td>
<td>11 (11)</td>
<td>5 (5)</td>
<td>3 (3)</td>
<td>5 (5)</td>
<td>44</td>
<td>45</td>
<td>97.80%</td>
</tr>
<tr>
<td>Filter 2</td>
<td>2 (3)</td>
<td>3 (4)</td>
<td>5 (5)</td>
<td>6 (7)</td>
<td>5 (7)</td>
<td>0 (1)</td>
<td>6 (11)</td>
<td>2 (3)</td>
<td>29</td>
<td>41</td>
<td>70.70%</td>
</tr>
<tr>
<td>Filter 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 (1)</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>100.00%</td>
</tr>
<tr>
<td>Filter 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 (2)</td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>33.30%</td>
</tr>
<tr>
<td>Filter 5</td>
<td>0 (1)</td>
<td>3 (5)</td>
<td>2 (3)</td>
<td>0 (0)</td>
<td>2 (2)</td>
<td>2 (4)</td>
<td>0 (0)</td>
<td></td>
<td>9</td>
<td>15</td>
<td>60.00%</td>
</tr>
<tr>
<td>Filter 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 (1)</td>
<td></td>
<td></td>
<td>12</td>
<td>13</td>
<td>92.30%</td>
</tr>
<tr>
<td>Filter 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 (9)</td>
<td>5 (8)</td>
<td>1 (4)</td>
<td>5 (7)</td>
<td>7 (10)</td>
<td>23</td>
</tr>
<tr>
<td>Filter 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
<td>75.00%</td>
</tr>
<tr>
<td>Filter 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 (2)</td>
<td>4 (5)</td>
<td>3 (3)</td>
<td>3 (3)</td>
<td>4 (5)</td>
<td>4 (5)</td>
</tr>
<tr>
<td>Filter 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 (2)</td>
<td>5 (5)</td>
<td>8 (9)</td>
<td>3 (5)</td>
<td>1 (1)</td>
<td>6 (7)</td>
</tr>
<tr>
<td>Exceptions</td>
<td>6</td>
<td>9</td>
<td>24</td>
<td>38</td>
<td>40</td>
<td>21</td>
<td>31</td>
<td>33</td>
<td>166</td>
<td>212</td>
<td>78.30%+</td>
</tr>
<tr>
<td>Errors Caught</td>
<td>5</td>
<td>7</td>
<td>20</td>
<td>30</td>
<td>32</td>
<td>15</td>
<td>20</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Errors Caught (%)</td>
<td>33.30%</td>
<td>26.90%</td>
<td>76.90%</td>
<td>85.70%</td>
<td>100.00%</td>
<td>68.20%</td>
<td>74.10%</td>
<td>82.40%</td>
<td>72.40%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>83.30%</td>
<td>77.80%</td>
<td>83.30%</td>
<td>78.90%</td>
<td>80.00%</td>
<td>71.40%</td>
<td>64.50%</td>
<td>84.80%</td>
<td>77.70%+</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

+ The discrepancy in Accuracy is due to some errors being marked as exceptions by several filters
Figure 28 describes the proportions of errors caught by the model and the rate of the 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 by 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 28. The portion of the errors caught and the false positive rate of exception reported by the model.](image)

**Factors Affecting the Performance 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.

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, 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).

Summary

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 audit framework, 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 the 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 chapter of my dissertation, I develop a data-driven model that can help to identify relevant filters for confirmed instances of reported exceptions and confirmed
outliers. The main advantage of analyzing confirmed outliers is deriving information about missing rules from the 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.

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 exception 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 risk factor 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, the 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 means of generating and improving rules from identified misstatements.
CONCLUSIONS

As more powerful data processing technologies and storage options became available, the continuous auditing framework has evolved from rule-based into data analytics based auditing through analyses of exceptions, outliers and anomalies. Data analytics methods serve as a complementary function by directing the auditor’s attention to the audit risks not addressed by the transaction verification filters. While the transaction verification module can be perceived as the auditor’s risk awareness, the outliers and anomaly detection at different aggregate levels is to identify the unusual items missed by filters and analyze their data values. A reliable continuous auditing model would have an accurate set of filters in transaction verification to catch intended exceptions and incorporate robust data analytics methods to minimize the possible audit risk not addressed in the transaction verification layer. In this dissertation, I review the existing literature on the implementation of continuous auditing on transaction-level data and discuss the issues in current models. To address these issues, I propose changes or additions to certain elements of the continuous auditing framework.

The first chapter of this dissertation discusses and evaluates the various options of the aggregation methods for rule-based risk filters from the distance metric perspective of the data analytics domain. I propose constructing a “risk space” from the filter values, assigning each filter of the transaction verification to a separate axis of the space. The distance between the “risk position” of an exception in the “risk space” and the origin represent the aggregated risk score for this exception. I consider the first two order variations of the Minkowski Distance (Manhattan and Euclidian) and the Mahalanobis Distance as the primary candidates among other distance metrics. I evaluate these metrics
by implementing them on the output of seven filters developed for a payroll dataset as a transaction verification module. The analyses revealed that most differences in ranking of exceptions are observed between any order of Minkowski Distance and the Mahalanobis Distance. In the same setting, the horizontal variance plays a major role in the differences among the orders of Minkowski Distance, lower-order Minkowski Distance favoring the exceptions with a lower horizontal variance. Moreover, Mahalanobis Distance allows the adjustments for the relationships among filter values, as filters assessing similar risk factors may inflate the risk scores of some exceptions. These findings may serve auditors in properly balancing the contribution of the filters to the risk score.

In the second chapter, I argue that the cross-similarity comparison of selected exceptions in the transaction verification must be introduced to the continuous auditing models for the efficient use of audit resources. To support this argument, I presented and analyzed the outcomes of the audit selection from synthetic and real-life datasets at various similarity thresholds, including the zero similarity threshold, which forces the algorithm to ignore the similarity among selected exceptions. If the similarity during the exceptions selection is ignored, some of the selection set members may represent the same audit risk factor, resulting in suboptimal use of limited audit resources. To avoid this problem, I propose two algorithms, the Skipper and the Stretcher, each aiming to achieve one of two selection objectives: risk maximization and similarity minimization among selected exceptions. Algorithms were tested on synthetic and real-life datasets for their robustness and invariance to changes in shapes and densities of the datasets.

The third chapter proposes a rule-induction based feedback loop from confirmed misstatements caught as an outcome of transaction verification and outlier detection in a
transaction-level continuous audit. This feedback loop evaluates the existing filters of the transaction verification and recommends new filter thresholds, if necessary. Moreover, if a certain high-risk subspace was reported by outliers but not by exceptions, the proposed model suggests a new filter to be added to the transaction verification by inducing a new rule for the uncovered novel misstatements. The feedback loop from confirmed outliers points auditor’s attention to the subspaces that are confirmed to be risky but were not addressed by the filters of the transaction verification. This model was evaluated on a nine-month payroll dataset, taking each month as a separate period. Evaluations showed that user-provided parameters (rule complexity, class ratio and minimum class instances) are significant factors to execute the model properly. Additionally, the performance of the feedback loop in inducing appropriate filters for future periods depends on the availability of sufficient data to learn from and the misstatements of future period data following historical patterns.

The artifacts proposed in the chapters of this dissertation introduce various extensions to the different components of the continuous audit framework. The methods represented by the proposed artifacts serve to increase accuracy through more informed risk scores (Chapter 1), achieve better efficiency through selecting diverse risk items (Chapter 2), and enhance the feedback exchange between the verification and analytical layers of the framework (Chapter 3). While proposed artifacts amend specific existing methods or add new techniques in different components of the continuous audit framework, these models complement one another and contribute to the contemporary evolution of the framework. Moreover, the artifacts from the first two chapters can also be implemented to significantly improve the performance of the MADS of No et al. (2019).
The evaluations of the artifacts proposed in this dissertation were carried out on a real-life payroll dataset. The value of these methods to the audit profession is revealed when they are implemented in actual audit settings. Eulerich and Kalinichenko (2018) review the literature on continuous audit and argue that, despite the significant research efforts devoted to developing theoretical guidance for the implementation, there are substantial research needs and opportunities to assess the effectiveness and efficiency of continuous audit in actual audit settings. Furthermore, the authors also find a lack of empirical evidence on cost-benefit analyses and the impact of the framework implementation on publicly recognized factors: quality of reported earnings, external audit fees, behavior of stakeholders, etc. Future research could investigate how each method proposed in this dissertation affects the audit procedures and environment in actual audit settings.

In this dissertation, I contribute to the literature by assessing the risk scores from the distance metric perspective, incorporating similarity measures in the selection of items for examination, and analyzing the confirmed instances for rule induction. Evaluations of the proposed artifacts through implementations on a real-life dataset proved the importance of the auditor’s knowledge about recordkeeping practices and the dataset distributions. Auditors are advised to familiarize themselves with the dataset using descriptive statistics and data analytics tools before implementing proposed models of the continuous audit.
REFERENCES


# APPENDIX A: DATA DICTIONARY FOR A PAYROLL DATASET

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

Table 20. Information about the fields of the payroll dataset used for the evaluation of the models.
APPENDIX B: HOW VARIANCES AND COVARIANCES AFFECT MAHALANOBIS DISTANCE

Sherman and Morrison (1950) offer an updating method of inverse matrix when a single element changes its value in given matrix. Following inside the rectangle is excerpt (screen cut) from Sherman and Morrison (1950).

\[
a_{ij}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n \text{ denote the elements of an } n \times n \text{ square matrix } a; \\
b_{ij}, \text{ denote the elements of } b, \text{ the inverse of } a; \\
A_{ij}, \text{ denote the elements of } A \text{ which differs from } a \text{ only in one element, say } A_{rs}; \\
B_{ij}, \text{ denote the elements of } B, \text{ the inverse } A.
\]

Let

\[
A_{rs} = a_{rs} + \Delta a_{rs}.
\]

The set of equations by means of which \( B \) may be computed from \( \Delta a_{rs} \) and \( b \) is

\[
(1) \quad B_{ij} = b_{ij} - \frac{b_{ir} b_{jr} \Delta a_{rs}}{1 + b_{ir} \Delta a_{rs}}, \quad r = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n,
\]

provided that \( 1 + b_{ir} \Delta a_{rs} \neq 0 \).

Using the formula above, the effects of variances and covariances among filter values can be calculated.

**How do variances filter values affect Suspicion Score?**

From the formula above, it can be deduced that if the covariance matrix of filter values is an identity matrix, i.e. there is no correlation between filter values and variance of each filter is 1, Mahalanobis Distance reduces to Euclidean Distance as the inverse of an identity matrix is itself. If the covariances matrix is diagonal, i.e. all pairwise covariances are 0, but variances are other than 0 and 1, then Mahalanobis Distance is Euclidean Distance adjusted for variances. This can be shown as the inverse of a diagonal matrix is
also a diagonal matrix with diagonal values equal to reciprocals of the corresponding diagonal values in the original matrix.

\[
\begin{array}{cccccc}
v1 & 0 & 0 & 0 & \ldots & \\
0 & v2 & 0 & 0 & \ldots & \\
0 & 0 & v3 & 0 & \ldots & \\
0 & 0 & 0 & v4 & \ldots & \\
\ldots & \ldots & \ldots & \ldots & \ldots & \\
\end{array}
\]

\[
\begin{array}{cccccc}
1/v1 & 0 & 0 & 0 & \ldots & \\
0 & 1/v2 & 0 & 0 & \ldots & \\
0 & 0 & 1/v3 & 0 & \ldots & \\
0 & 0 & 0 & 1/v4 & \ldots & \\
\ldots & \ldots & \ldots & \ldots & \ldots & \\
\end{array}
\]

**Figure 29. Inversion of a diagonal matrix.**

\[
MD = \sqrt{\frac{x_1^2}{v_1} + \frac{x_2^2}{v_2} + \frac{x_3^2}{v_3} + \ldots}
\]

*How does covariance between two filter values affect Risk Score?*

\[
\begin{array}{cccccc}
v1 & 0 & 0 & 0 & \ldots & \\
0 & v2 & 0 & c42 & \ldots & \\
0 & 0 & v3 & 0 & \ldots & \\
0 & c42 & 0 & v4 & \ldots & \\
\ldots & \ldots & \ldots & \ldots & \ldots & \\
\end{array}
\]

\[
\begin{array}{cccccc}
1/v1 & 0 & 0 & 0 & \ldots & \\
0 & D1 & 0 & T & \ldots & \\
0 & 0 & 1/v3 & 0 & \ldots & \\
0 & T & 0 & D2 & \ldots & \\
\ldots & \ldots & \ldots & \ldots & \ldots & \\
\end{array}
\]

**Figure 30. Inversion of a diagonal matrix.**

\[
D1 = \frac{v_y}{v_x v_y - cov_{xy}^2} = \frac{v_y/v_x v_y}{(v_x v_y - cov_{xy}^2)/v_x v_y} = \frac{1}{v_x(1 - cor_{xy}^2)}
\]

\[
D2 = \frac{v_x}{v_x v_y - cov_{xy}^2} = \frac{v_x/v_x v_y}{(v_x v_y - cov_{xy}^2)/v_x v_y} = \frac{1}{v_y(1 - cor_{xy}^2)}
\]

\[
T = \frac{-cov_{xy}}{v_x v_y - cov_{xy}^2} = \frac{-cov_{xy}/v_x v_y}{(v_x v_y - cov_{xy}^2)/v_x v_y} = \frac{-cor_{xy}}{\sqrt{v_x v_y(1 - cor_{xy}^2)}}
\]
Figure 31. Separate effects of variance and covariance on Mahalanobis Distance
APPENDIX C: DECOMPOSITION OF COVARIANCE MATRIX

Figure 32. Decomposition of Mahalanobis Distance to variance and covariance effects.
APPENDIX D: GREEDY HEURISTIC APPROACH BASED A RECURSIVE RULE INDUCTION

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

Obtained Rule: Tuition \( \geq 75.0 \). 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 33. First Rule Induction Run for Period 1.

Figure 34. Second Rule Induction Run for Period 1.
**Obtained Rule:** Overtime \( \geq 571.38 \). This rule is simplest and matches the user parameters: maximum negative class ratio \( (\text{max}\_\text{class}\_\text{ratio} = 1) \), minimum positive instances \( (\text{min}\_\text{positives} = 3) \) 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.

![Decision Tree Diagram](image)

*Figure 35. Third Rule Induction Run for Period 1.*

No rule can be obtained that matches user parameters: \( \text{max}\_\text{depth} = 3 \), \( \text{max}\_\text{class}\_\text{ratio} = 1 \) and \( \text{min}\_\text{positives} = 3 \).