UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Unsupervised Outlier Detection in Financial Statement Audits

Rick J. Lenderink

M.Sc. Thesis

Business Information Technology (BIT)

September 2019

Supervisors:
dr. M. Poel
prof. dr. J. van Hillegersberg

University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Acknowledgements

The writing of this thesis has been an exciting and interesting journey. During my internship at de Jong & Laan and in relation to this research, I met many remarkable people who taught me about the field of accounting and financial statement audits. I would like to express my sincere gratitude to them and everyone who helped me arrive at my final destination - the successful completion of this graduation project.

First of all, I would like to thank Mannes Poel, Guido van Capelleveen and Jos van Hillegersberg, for guiding me in this thesis writing process and all the valuable feedback. Your feedback during the meetings has been very useful, motivational and kept me on the right track in order to write my thesis. During the meetings innovative ideas have inspired me in order to improve the quality of my research at de Jong & Laan.

I thank Dennis Krukkert, Jasper Visserman and Thijs de Nijs for their direct support during my research period. You provided me with tools and great feedback which ensured that I remained critical and always kept learning new things. Also I like to thank my direct colleagues, my time with you has been very pleasant and you made sure my days at de Jong & Laan have been very enjoyable.

Thank you!

Rick Lenderink

Vroomshoop
September 30, 2019

Summary

During financial statement audits great amounts of transactional data is examined by auditing accountants to provide assurance that an organization's financial statements are reported in accordance with relevant accounting principles. This thesis focuses on the application unsupervised outlier detection techniques to aid auditors in finding outlying journal entries that could be of interest in terms of fraud or errors made. In order to conduct fraud, one has to deviate from 'normal' behaviour and from regular financial transaction patterns. The same can be said about errors made in financial administrations, erroneous transactions are rare and deviate from a regular transaction pattern. It is therefore believed that there is a link between abnormal journal entries (outliers / anomalies) and financial fraud or financial errors.

Based on systematic literature research unsupervised outlier detection techniques are categorized in: proximity-based techniques, subspace techniques and statistical / probabilistic models. Alongside there are unsupervised outlier detection techniques that do not reside in any of these three categories and are rather a technique on their own. In total 11 unsupervised outlier detection techniques have been listed and categorized. From these techniques, Isolation Forests (IF), K-Nearest Neighbors (KNN), Histogram-based Outlier Score (HBOS) and Autoencoder Neural Networks has been selected in order to conduct experiments with. Based on previous literature these four techniques seem to be most promising in order to detect outliers in transactional audit data sets.

The four techniques have initially been experimented with on a realistic transactional audit data set consisting of $4,879$ journal entries. Together with certified public accountants of de Jong & Laan, synthetic outlying journal entries have been inserted in this data set. 7 synthetic outliers have been injected in the data set from which 5 are global outliers and 2 are local outliers. In total a proportion of 0.14% synthetic outliers are therefore known and labeled as outlier, turning it into a partially supervised problem.

The selected techniques are evaluated based on their detection rate of the synthetic outliers. All outlier detection techniques have an outlier score as output, providing each journal entry with an outlier score. Performance is measured based on the proportion of top journal entries that have to be selected based on outlier score in order to obtain a recall of 100% for the synthetic outliers. In other words, sorting journal entries based on their outlier score, how many of these top scoring journal entries are to be included in order to contain all synthetic outliers. In case of Isolation Forests on average basis, only the top 2.12% of journal entries include all synthetic outlying journal entries. This makes Isolation Forest the

best performing outlier detection technique during these experiments. K-Nearest Neighbors scored a percentage of 19.31%, Histogram-based Outlier Score 3.54% and Autoencoder Neural Networks 56.78%.

Based on the results Isolation Forests is applied during two real audit cases, providing an outlier score to journal entries from two clients from de Jong & Laan Accountants. Based on a threshold, all journal entries having a higher outlier score than 0.6 have been examined by a certified public accountant and corresponding auditor from both clients. The accountant and auditor indicated for each journal entry whether they could be of interest from an audit perspective (anomalous / non-anomalous). Alongside the auditor indicated for each journal entry whether they have been detected to some extend during the regular audit process or not.

For the first client this resulted in 150 journal entries that have been examined from which $53(35.33\%)$ were labeled as anomalous by the auditor and from which $4(2.97\%)$ have not been detected during the regular audit process. For the other client, 51 journal entries were examined from which $13(25.49\%)$ have been labeled as anomalous by the auditor and $3(5.88\%)$ were undetected during the audit.

This concludes that unsupervised outlier detection techniques and more specific, Isolation Forests, are suitable in order to detect outliers that are of interest during financial statement audits. Isolation Forests has been able to provide auditors with abnormal journal entries that haven't been detected following regular audit procedures. Applying these techniques therefore reduce the risk of missing anomalous journal entries that could be of interest and so improves the quality of financial statement audits.

Contents

List of Figures

List of Tables

List of Acronyms

ACFE	Association of Certified Fraud Examiners
Adam	Adaptive moment estimation
AIS	Accounting Information System(s)
CPA	Certified Public Accountant
HBOS	Histogram-Based Outlier Score
IF	Isolation Forests
IFRS	International Financial Reporting Standards
KNN	K-Nearest Neighbor
LOF	Local Outlier Factor
LOCI	Local Correlation Integral
LoOP	Local Outlier Probability
LReLU	Leaky Rectified Linear Units
MSE	Mean Squared Error
NBA	Koninklijke Nederlandse Beroepsorganisatie van Accountants
PCA	Principal Component Analysis
SVM	Support Vector Machines
XAF	XML Auditfile Financieel

Chapter 1

Introduction

According to the Association of Certified Fraud Examiners (ACFE), financial statement fraud accounts for about 10% of white-collar crime. The research done in 2018 by ACFE where over 2.690 real cases of occupational fraud from 125 countries in 23 industry categories have been analyzed, states that financial statement fraud is the least common but most costly type of fraud [4]. ACFE describes occupational fraud as fraud committed against the organization by its own officers, directors, or employees. Globally, the total loss of occupational fraud is estimated to be more than $ 7 billion with a median loss of $ 130.000 per case [4]. Focusing on financial statement fraud cases (10%), we'll find that these have a median loss of $ 800.000.

The Koninklijke Nederlandse Beroepsorganisatie van Accountants (NBA), the professional body for accountants in the Netherlands is, among other things, responsible for is promoting proper professional practice of its members (chartered accountants). Accountants have an important role in the prevention of fraud by performing financial statement audits on medium- and large-sized companies. The purpose of a financial audit is to ensure that financial information - such as the financial statements - does not contain material misstatements that are the result of fraud or errors. In their fraud protocol the NBA describes fraud as "deliberate deception to obtain an unlawful advantage" [5].

Machine learning techniques could aid the auditor in ensuring that financial information is correct. A promising direction is unsupervised outlier detection or unsupervised anomaly detection, which could make the audit process more effective and more efficient by analyzing transactional audit data. Outlier detection techniques intent to detect 'abnormal' instances, being observations that deviate markedly from the rest. These outliers could be a possible indication for errors or fraud in transactional data sets, therefore detecting outliers could aid financial statement auditors.

Overall there has been an increase in the use of analytical procedures, including machine learning techniques, in external auditing and many research has been done to the use of analytical techniques in external audits. Academia has already conducted extensive research regarding the use of expanded analytics in the external audit, yet even more is required [6].

1.1 Unsupervised Outlier Detection

Finding or detecting *"not-normal"* instances is the process of anomaly detection or outlier detection. The field of unsupervised outlier detection focuses on finding outliers or anomalies in data sets in an unsupervised manner. One of the first definitions of an outlier is given by Grubbs [7]: *"An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs"*. This definition is extended by Goldstein and Uchida [3] with two important characteristics of an anomaly being: *"anomalies are different from the norm with respect to their features, and they are rare in a data set compared to normal instances"*.

Outlier detection methods can be divided between supervised and unsupervised methods [8]. In case of supervised it means that an outlier detection model is trained based on a data set where the outliers of that data set are known. Supervised outlier detection can be considered as imbalanced classification problems (since the class of outliers has inherently relatively few members) [8]. The trained models can be applied on new unseen data sets in order to classify outliers.

Unsupervised outlier detection techniques train models based on data sets where the outliers are unknown, meaning that characteristics of an outlier are unidentified. The idea is that an unsupervised anomaly detection method scores data solely based on intrinsic properties of the data set [3]. Outlier detection is generally considered as an unsupervised learning challenge due to lack of prior knowledge about the nature of various outlier instances [9]. Moreover, unlabeled data is available in abundance and obtaining labeled data is expensive in many scenarios as with the case of de Jong & Laan Accountants.

An example of traditional unsupervised outlier detection techniques are proximity-based techniques like K-Nearest Neighbor (KNN). Proximity-based techniques define a data point as an outlier when its locality is sparsely populated according to [10]. Other techniques are subspace techniques like principal component analysis, and statistical / probabilistic models like mixture modelling. Furthermore there has been an increase in the application of Neural Networks and deep learning models, Neural Networks in the form of an Autoencoder can be applied to detect outliers based on reconstruction error.

1.2 Problem Statement

As companies grow more complex, automated systems and processes have necessarily become much more prevalent. Accounting Information System(s) (AIS) are one of these automated systems, tracking financial streams and processes, and storing an increasing amount of data within companies [6], [11]. Financial data from medium and large-sized companies are lawfully required to be analyzed and assessed each year by an external accountant in the form of a financial statement audits. A financial statement audit provides an audit opinion as a result. A positive opinion indicates that reasonable assurance has been obtained that the financial statements are free from material misstatement, whether due to fraud or error, and that they are fairly presented in accordance with the relevant accounting

standards (e.g. International Financial Reporting Standards (IFRS)) [12]. Misstatements are material if they could, individually or collectively, influence the economic decisions that users (e.g. stakeholders) make on the basis of the financial statements. An audit opinion does not provide any guarantee, but it is rather a statement of professional judgement. The auditor cannot obtain the absolute assurance that financial statements are free from material misstatement simply because not everything can be analyzed during financial statement audits and there are certain limitations. The auditor obtains reasonable assurance by gathering evidence through selective testing of financial records.

During financial statement audits great amounts of data are analyzed. Use is made of a standardized data sets, 'audit file', containing all financial transactions (journal entries) from a specific company in order to draft the annual financial report. Analyzing this financial audit data is increasing in complexity and a time-consuming task for the auditor. Unsupervised outlier detection techniques can very well aid the auditor in finding outlying or abnormal transactions in these data sets. It is believed that in order to conduct fraud, a perpetrator must deviate from regular financial transaction patterns. Deviations are recorded by a small number of journal entries, therefore making these journal entries abnormal or "anomalous" / "outlying". The same is to be said about errors made in financial administration systems, erroneous transactions could deviate from normal behaviour. In case of transactional data sets, outlying journal entries could be found automatically utilizing outlier detection techniques therefore supporting the auditor during audits. Unsupervised outlier detection techniques can analyse great amounts of data in a relative short amount of time so it has the potential to make financial statement audits more efficient. Besides being able to analyze entire data sets with an outlier detection technique could reduces the risk of missing any abnormalities that could be an indication of fraud or errors made. The potential of the application of unsupervised outlier detection techniques on financial audit data is unclear so far since few academic research is done in equal problem setting.

1.3 Research questions

Based on the the introduction and problem statement the following main research question has been formulated for this thesis:

To what extend can unsupervised outlier detection techniques be applied to detect outliers in transactional audit data?

Sub research questions:

RQ1 *Which (class of) unsupervised outlier detection techniques can be applied on transactional audit data in order to detect outliers?*

RQ2 *Which unsupervised outlier detection technique performs best in detecting outliers that are of interest for the auditor?*

1.4 Report organization

This thesis is structured as following:

- *Chapter 2;* Describes relevant background information about financial statement audits and audit data sets. Besides, specific unsupervised outlier detection algorithms experimented with during this research shortly explained. These techniques are: Isolation Forests (IF), Autoencoder Neural Network, Histogram-Based Outlier Score (HBOS) and K-Nearest Neighbor (KNN)
- *Chapter 3;* Contains relevant academic literature available in the context of unsupervised outlier detection techniques. Results are summarized in the form of a table presenting potential unsupervised outlier detection techniques applicable in the context of financial statement audits
- *Chapter 4;* Describes the research methodology utilized during multiple case studies. Starting from model selection, data preprocessing to experimenting
- *Chapter 5;* Describes the setup process of initial experiments, four unsupervised outlier detection methods are selected and a realistic audit data set is prepared. Based on injected synthetic outliers the selected unsupervised outlier detection techniques are compared based on detection performance
- *Chapter 6;* Describes single case experiments where four different unsupervised outlier detection technique are evaluated based on realistic transactional audit data containing synthetic injected outliers
- *Chapter 7;* Application of the unsupervised outlier detection technique Isolation Forests (IF) is described. IF is applied on transactional audit data sets from two clients of de Jong & Laan and evaluated with support of three audit experts
- *Chapter 8;* Discusses the implications of the results and evaluates validity, reliability and limitations
- *Chapter 9;* Concludes this thesis and directly answers the proposed research questions

Chapter 2

Background

This chapter describes relevant background information in order to enable a better understanding of this study. Financial statement audits will be described in general along with existing data analysis methods utilized on transactional audit data. This will be followed by a description of unsupervised outlier detection techniques that have been used during multiple experiments in this study.

2.1 Financial Statement Audits

A financial statement audit is an independent and objective evaluation of an organization's financial reports and financial reporting processes. An organization produces financial statements to provide information about their financial position and performance. This information is used by a variety of stakeholders (e.g. investors, banks, suppliers) in making economic decisions. The financial statement audit is a service provided by accounting firms. Companies, small, medium and large-sized, are legally required to publish their financial statements annually. The financial statements of medium and large size companies require to be audited by an independent qualified external auditor. The primary goal for financial audits is to provide stakeholders reasonable assurance that financial statements are accurate and complete. According to a report released by PricewaterhouseCoopers an audit consists of an evaluation of a subject matter with a view to express an opinion on whether the subject matter is fairly presented [12]. To provide a fair representation of reality a companies require an audit report before the annual financial statements are published.

The auditor's report is the result of the audit and provides a high level of certainty about the financial performance and therefore trust among stakeholders. The report expresses an opinion indicating that reasonable assurance has been obtained that the financial statements are free from material misstatement, whether due to fraud or error [12]. Furthermore the opinion indicated that the financial statements are fairly presented in accordance with the IFRS. It is the auditors responsibility to plan and conduct the audit in such a way that it meets the auditing standards and sufficient appropriate evidence is obtained to support the audit opinion. However, what constitutes sufficient appropriate evidence is a matter of professional judgement and experience. An auditor gives a 'clean' opinion when it is

concluded that financial statements are free from material misstatement. Misstatements are material if they could, individually or collectively, influence the economic decisions that users make on the basis of the financial statements according to an article posted on the website of the IFRS [13].

Figure 2.1: Phases of a financial statement audit

Figure 2.1 represents an overview of the multiple phases during a financial statement audit. The first two phases, preparation, planning and exploration leading up to the audit assignment and focus on the acceptance of the client by the audit firm. During the third phase, strategy and risk estimation, auditors use their knowledge of the business, industry and environment in which a company operates in order to identify risks. Based on the risks a detailed audit plan is generated in order to address the risks of material misstatement in the financial statements. This includes a testing approach to various financial statement items. During audit execution an auditor gathers information trough a combinations of testing the company's internal controls, tracing amounts and disclosures included in the financial statements to the client's supporting books and records, and obtaining external third party documentation. Independent confirmation may be sought for certain material balances such as cash position. Substantive testing procedure during audit execution can include:

- *Inspecting physical assets such as inventory or property*
- *Examining records to support balances and transactions*
- *Obtaining confirmation from third parties such as suppliers and customers*
- *Checking elements of the financial statements such as a price comparison based on external market indexes*

Finally, during conclusion and finalisation, a conclusion is formed and the auditor provides the audit opinion in the auditors report.

2.1.1 Audit Execution: Data Analysis

As part of the audit execution phase data sets are analyzed. The auditor uses of multiple techniques to analyze the data looking for anomalies that could possibly be caused deliberately or by mistake. The detection of anomalies in transactional data sets can be an incredibly difficult task [14]. In a study by Appelbaum et al. [6] it is shown that the majority of techniques used during an audit process are: ratio analysis, transaction tests, sampling, data modeling and data analytics. Indicating that machine learning techniques are not widely adopted by practitioners in the field of financial auditing.

Transactional data sets included in financial audits are extracted from an AIS. They include all information, from a certain financial year, required to generate a financial statement of an organization. Included are general ledger accounts, daybooks, journal entries, bank accounts, tax details, customers and possible sub-administrations. General ledger accounts store transactions which represent the customer's general ledger, they can be divided into two separate groups, namely: balance sheet accounts and income statement accounts. Daybooks contain a chronological order of transactions that take place within a company, these daybooks contain journal entries. A journal entry describes a single financial transaction within a financial administration by describing a debit and a credit balance. Furthermore journal entries can possibly be linked to a customer or a supplier. The journal entries are lowest level of abstraction in a financial data set and provide detail about a single transaction. The journal entries are fundamental in order to create and publish a yearly financial statement, and therefore fundamental in the auditing process.

A technique of analyzing transactional data, widely applied for financial audits, is Benford's Law [11]. Benford's law is applied in the economic world in order to detect irregularities and possibly fraud in financial data [15]. Benford found that many numerical data sets do not follow a uniform distribution for the first digit, as one might expect [16]. Instead these first digits follow a different distribution presented in table 2.1:

First Digit	1	2	3	4	5	6	7	8	9
Probability	0,3010	0,1761	0,1249	0,0969	0,0792	0,0669	0,0580	0,0512	0,0458

Table 2.1: Benfords Law: Distribution for first digit in numerical data sets

In case of this research the focus lies on finding outliers in the audit data sets that could possibly indicate fraud or errors utilizing unsupervised outlier detection algorithms. It is up to the auditor to decide whether an outlier is an indication of fraud.

2.1.2 Transactional Data

Transactional data describe an internal or external event or transaction that takes place as an organization conducts its business [17]. Examples of transactional data are sales orders, invoices, purchase orders, shipping documents, credit card payments etc. A transactional data set consists of a number of transactions, each of which contains a varying number of items [18]. Furthermore transactional data is particularly facet of categorical data [18].

The described outlier detection techniques in chapter 3 have been applied on many different data sets and not specifically transactional data sets. From the described techniques only a single technique, Autoencoder Neural Network [2], has been experimented with extensively on the same type of transactional data (journal entries) and with the same purpose as this study. The research also applied other outlier techniques on the transactional data set in order to evaluate the performance of an Autoencoder Neural Network. The techniques that have also been applied are: Local Outlier Factor, One-class Support Vector Machine, Principal Component Analysis and Density-Based Spatial Clustering of Applications with

Noise [2]. No other techniques have been found in the literature that have also been experimented with thoroughly on transactional audit data consisting of journal entries.

Specific for this research transactional data consist out of journal entries describing a financial event in an organization. The journal entries also contain mainly categorical variables. Data sets utilized during this research are also used during financial audit statements of de Jong & Laan. These data sets are extracted from client AIS, as described before and are called 'XAF audit files'. XML Auditfile Financieel (XAF) is a standardized format developed in the Netherlands by the SRA (samenwerkende registeraccountants & accountants-administratieconsulenten) and the dutch tax authorities [19]. The SRA is the collaboration of accountants in the Netherlands, one of their goals being to improve overall quality of accounting. A XAF file has an identical structure as XML and the current structure (XAF v3.2) is presented in appendix A.

2.2 Unsupervised Outlier Detection Techniques

The unsupervised outlier detection techniques utilized during this research will be elaborated shortly in this section. Starting with Isolation Forests (IF), the algorithm that has been used most extensively. Followed by Autoencoder Neural Networks, Histogram-Based Outlier Score (HBOS) and K-Nearest Neighbor.

2.2.1 Isolation Forests (IF)

Lui, Ting & Zhou propose a tree-based unsupervised outlier detection technique named 'Isolation Forests' (IF) [20]. Zhao, Nasrullah & Li implemented this this technique in a Python package, which has been used during this study [21], who among Isolation Forests implemented multiple outlier detection techniques in a Python library.

Isolation Forest shares intuitive similarity with another tree-based algorithm called 'random forest' [10], which is mainly used for classification problems. In a book about outlier detection, Aggarwal describes IF as an ensemble combination of a set of isolation trees (estimators) [10]. Lui describes the term 'isolation' as *'separating an instance from the rest of the instances'*. In a single isolation tree, the data is recursively partitioned with axis-parallel cuts at randomly chosen partition points in randomly selected attributes (features). This is done for n data points to isolate the points into nodes with fewer and fewer points until they are isolated in singleton nodes containing one instance [10]. The intuition behind the technique is that tree branches containing outliers are noticeably less deep, because these data points are located in sparse locations. The distance of the leaf to the root is used as the outlier score. Since IF creates multiple trees (n estimators) the average path length for each data point is calculated over the different trees in the isolation forest. IF functions under the assumption that it is more likely to be able to isolate outliers. Hence, when a forest of random trees collectively produce shorter path lengths for some particular points, they are likely to be anomalous [20].

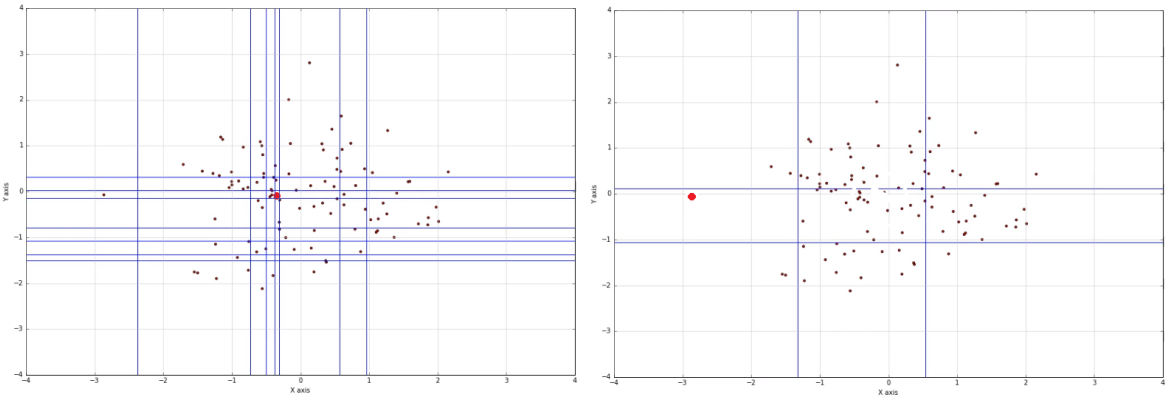

Figure 2.2: Isolation process of two data points in a two dimensional setting. On the left random splittings (blue lines) are visualized to isolate a 'normal' data point. On the right an 'anomalous' data point is visualized requiring less random splits. Source: [1]

Figure 2.2 visualizes the isolation process for two data points. Here it is clearly visible a 'normal' data point requires a lot more splits (path length = 14) on both attributes of the data in order to isolate the data point. On the right side of the figure the data is only partitioned four times (path length = 4) in order to isolate the 'anomalous' data point. In each isolation tree *n* data points are isolated in this way and again the average path length for each data point determines it's outlier score. For a more detailed explanation of Isolation Forests see the paper of Lui [20].

2.2.2 Autoencoder Neural Network

Schreyer, Sattarov & Borth applied an Autoencoder Neural Network on two data sets containing journal entries in order to detect outliers based on the reconstruction error for each reconstructed journal entry. An Autoencoder or Replicator Neural Network is a special type of feed forward multilayer neural network that can be trained to reconstruct its input [2]. The difference between the original input and its reconstruction is referred to as reconstruction error. An overview of an Autoencoder Neural Network is visualized in figure 2.3.

During this study the research of Schreyer et al. has been fundamental in order to develop Autoencoder Networks, meaning that the same architecture and hyperparameters have been used initially whereas changes has been made in order to make improvements. For more details see E.

Following the paper of Schreyer et al. Autoencoder Neural Networks compromise two non-linear mapping referred to as an encoder and a decoder. Usually these have a symmetrical architecture consisting of several layers of neurons each followed by a non-linear function and shared parameters. The encoder maps an input vector x^i to a compressed representation z^i in the latent space Z [2]. The latent representation z^i (indicated red neurons figure 2.3) is then mapped back by the decoder to a reconstructed vector $\hat{x}^i$, being the reconstruction of the original input. The network learns by minimizing the dissimilarity of a

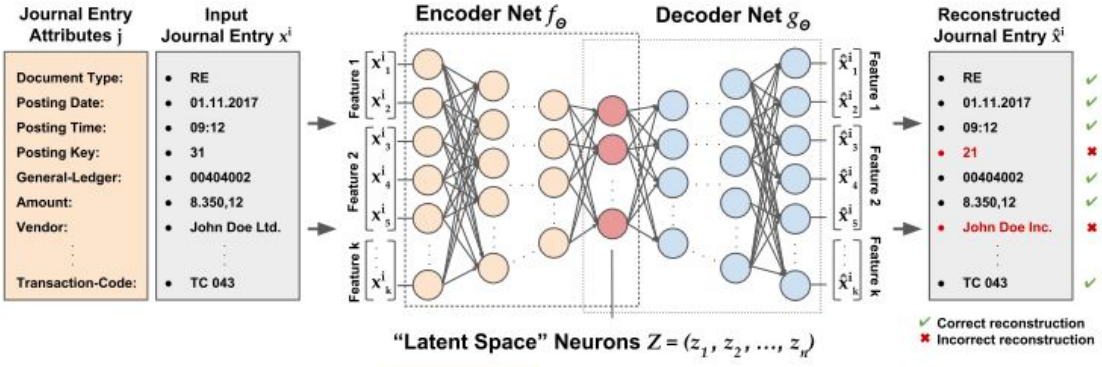

Figure 2.3: Schematic overview of an Autoencoder Neural Network reconstructing its input journal entry. A reconstruction error is calculated based on the difference between a reconstruction and the original input. Source: [2]

given journal entry x^i and reconstruction $\hat{x}^i$. The dissimilarity is calculated by a loss function. The loss function utilized during experiments conducted by Schreyer et al. is cross-entropy loss.

To prevent the Autoencoder from learning the identity function, the neurons in the hidden layers are reduced referred to as a ”bottleneck” [2]. Schreyer et al. indicate that such a constraint forces the Autoencoder to learn an optimal set of parameters that result in a compressed model of the most prevalent journal entry attribute value distributions and dependencies. The essence is that anomalous journal entries are harder to reconstruct due to their rarity and the neural network does not optimize for the rare instances. These anomalous journal entries thus have a higher reconstruction error than ’normal’ journal entries and are therefore scored more anomalous based on a higher reconstruction error.

2.2.3 Histogram-Based Outlier Score (HBOS)

During this research Histogram-Based Outlier Score (HBOS) as proposed by Goldstein & Dengel [22] has been utilized during experiments. The implementation from RapidMiner Studio [23] has been used during this research.

Goldstein describes HBOS as a simple statistical anomaly detection algorithm that assumes independence of features [3]. The idea being that for each feature in the data set a histogram is created. In case of categorical features, simple counting of the values of each category is performed and the relative frequency is computed. For numerical features, two different methods can be used in order to compute histograms, being static bin-width or dynamic bin-width histograms [22]. For each feature a histogram is created where the height of each single bin represents a density estimation. Histograms are then normalized so that the maximum height is 1.0, equaling the weight of each feature. The outlier score for each

data point x is calculated by the following formula [22].

$$HBOS(x) = \sum_{i=0}^{f} log(\frac{1}{hist_i(x)}) \tag{2.1}$$

Where f indicates the feature and $hist_i(x)$ indicates the height of the bin x resides. The idea, according to Goldstein, is very similar to Naive Bayes where independent feature probabilities are multiplied.

2.2.4 K-Nearest Neighbor (KNN)

KNN is likely the most well known technique used during this research. During this research use has also been made by the Python implementation of Zhao [21].

An outlier score of a data point is calculated by its average distance to its k nearest neighbors. According to Goldstein, as a rule of thumb, k would be in the range 10 $<$ k $<$ 50 [3]. The distance measure used during experiments described in this research is *euclidean* distance. Figure 2.4 visualized the results of KNN outlier detection technique in a two dimensional setting, whereas the circle size represents the anomaly score.

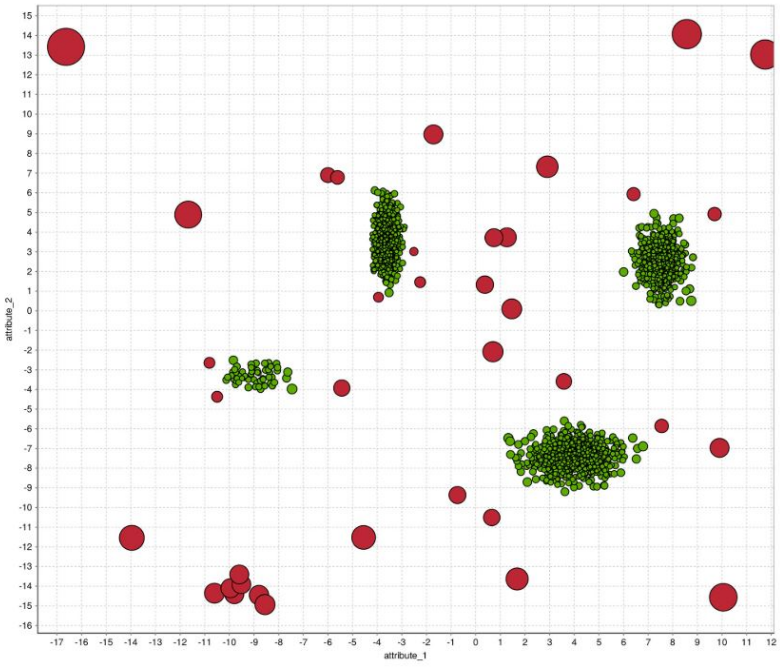

Figure 2.4: Visualization of KNN outlier detection algorithm output. The red points are anomalous whereas the circle size represents the anomaly score. Source: [3]

Chapter 3

State of the Art

This chapter describes the current state of the art of unsupervised outlier detection techniques, therefore a scientific literature review has been conducted. The research method of this literature review has been described in section 3.1 followed by multiple unsupervised outlier detection categories in section 3.2. Section 3.3 provides an overview of multiple unsupervised outlier detection techniques which is presented in table 3.1. Alongside, different types of outliers are described together with labeling and scoring outputs of unsupervised outlier detection techniques

3.1 Research Method

The main objective of this literature study is to explore and categorize available unsupervised outlier detection techniques that can be applied in a financial audit process. Systematic research is conducted following the procedure of Kitchenham [24] to organize existing research, select, categorize and shortly describe techniques that are proven to be successful. The final goal is to be able to conduct multiple experiments with outlier detection techniques in the form of case studies.

Keywords have been used in order to collect relevant research papers in Google Scholar but also from the following databases: SpringerLink, ScienceDirect, Scopus, IEEE Xplore, ACM Digital library and arXiv. Keywords included are: financial statement, financial auditing, external auditing, accounting, journal entry, fraud, data mining, data analytics, statistical analysis, algorithms, outlier detection, anomaly detection, machine learning, unsupervised, categorical data, mixed data. The keywords have been used to construct search strings i.e.: (outlier detection) AND (unsupervised) AND (financial auditing).

In order to determine whether a certain paper is included in this research a set of criteria has been created in order to perform a selection. Papers published in journals or online working papers are included. Papers should mention auditing, external auditing or financial auditing in combination with data analysis, statistical analysis, data mining, machine learning or algorithms are included. Papers reviewing and/or unsupervised outlier detection techniques are included, these papers can either be published in journals, conferences or working papers. A lot of papers technically describing a specific unsupervised outlier tech-

nique are conference papers, workshop papers or working papers are only included when a paper published in a journal references to one of these technical papers.

A couple of exceptions have been made since some of the unsupervised outlier detection techniques described in this research are mentioned in a research very specific to financial audits and prove to be successful. According to Garousi [25] it is important to include papers called ”grey literature”, which are non-peer reviewed papers, in software engineering. This is especially the case when there is not a substantial amount of literature available in a specific field of research, as with the case of unsupervised outlier detection applied specifically on financial audit data. State of the art concepts can be provided to the researcher when grey literature is included, furthermore according to Kitchenham it is important to include grey literature in order to reduce publication bias [24]. The exceptions, grey literature, included in this literature study come from research papers described very recently, are not peer-reviewed and do not have many citations, but the promising results in the field of financial auditing were decisive enough to include them in this research. Also a master thesis focusing on data-driven audit anomaly detection algorithms has been included in this research since similar specific research has been done in thesis. Furthermore two books published by Springer International Publishing have been included in this research. These books both describe unsupervised outlier detection techniques in high detail and are therefore included. Finally a recent paper by [21] has been a great source for unsupervised outlier detection algorithms. The paper substantiates a Python library for outlier detection called PyOD which is available since 2017.

Papers that do not discuss on any aspect of data analytics / statistical analysis / data mining / outlier detection / algorithms / machine learning are not included in this research. Incomplete and / or duplicate papers are not included. Papers that discuss specifically outlier detection techniques in a supervised or semi-supervised setting are not included. Papers describing specifically outlier detection techniques not applicable for transactional data are not included i.e. time series outlier detection or spatial data outlier detection. Papers only containing statistical methods are not included, based on the case description of de Jong & Laan these are of less interest.

3.2 Unsupervised Outlier Detection Categories

Common unsupervised anomaly detection techniques present in the current literature can be divided in multiple categories. Goldstein and Chandola [3], [26] roughly categorize unsupervised outlier detection techniques into: nearest-neighbor based, clustering-based, statistical and subspace techniques. De Wit [27] has the same approach for categorizing unsupervised outlier detection algorithms in his master thesis exploring the application of these techniques for data-driven audits. In a book about outlier analysis, Aggarwal categorizes outlier detection techniques in: probabilistic / statistical models, linear models, proximity-based, subspace methods and outlier ensembles [10]. All categories described thus far are roughly the same where Aggarwal describes nearest-neighbor based and clustering-based

techniques in a single category, namely proximity-based [10]. Aggarwal [10] categorizes Neural Networks, Support Vector Machines (SVM) and Principal Component Analysis (PCA) under linear models where Goldstein [3] categorizes SVM as 'other' and PCA as subspace technique. Outlier ensembles can be described as a combination of multiple outlier detection techniques where outputs of algorithms are combined. This research distinguishes the following categories where previous sources are combined:

- Proximity-based
  - *Distance-based*
  - *Clustering-based*
  - *Density-based*
- Statistical / probabilistic
- Subspace techniques
- Other

From these, proximity-based techniques are the most prominent for outlier detection [10]. Only taking the above techniques in account, proximity-based also occurs the most in a comprehensive literature from Appelbaum [6]. Proximity-based techniques define a data point as an outlier when its locality is sparsely populated according to Aggarwal [10]. Distance-based techniques measure the distance of a data point to its k-nearest neighbor. Data points with a large k-nearest neighbor distances are defined as outliers. In cluster-based techniques outliers can be quantified based on its distance from clusters, size of closest cluster or non-membership of a cluster. With density-based techniques one can find outliers based on the number of other data points within a specified local region. All these techniques are closely related based on the notion of proximity (or similarity) [10].

With statistical and probabilistic techniques the likelihood fit of a data point to a generative model is the outlier score [10].

Subspace techniques are primarily applied on high dimensional data sets. Data sets of de Jong & Laan can potentially become high dimensional since encoding of categorical data can increase the dimensions [2], [10]. With subspace techniques one detects subspaces in data sets, deviations from normal subspaces may indicate anomalous instances [3]. Furthermore algorithms such as support vector machines, Neural Networks and decision trees exist but do not fall into one of the previous described categories and are rather a specific technique on their own.

3.3 Unsupervised Outlier Detection Techniques, an Overview

This section provides an overview of machine learning algorithms. These algorithms have been selected based on recent papers about unsupervised outlier detection and are presented in table 3.1. Note that this is not an overview containing all possible algorithms in the field of unsupervised outlier detection but rather algorithms are proven to be useful [3], [28]. Distinction is made whether algorithms are suitable for categorical and/or numerical

attributes in data sets. Furthermore in order to give some representation of the popularity of the algorithm the number of citation has been given. These citation numbers are based on papers which technically describe a specific algorithm [21].

3.3.1 Outlier Types

Outlier can be divided into two types of anomalies, namely global anomalies and local anomalies. Data points that are very different from dense areas with respect to their attributes are called 'global anomalies' [3], [8], [10]. When a data point is only anomalous when compared to its close-by neighborhood it is called a 'local anomaly' [3], [8], [10]. Up until now an outlier is referred to as a single instance in the data set that deviate from the norm, this is called 'point anomaly detection' [26]. Nearly all available outlier detection algorithms are from this type [3]. Goldstein [3] describe a small cluster of multiple anomalous instances as a 'collective anomaly'. [3] further describe that point anomaly detection algorithms can be applied in order to detect collective anomalies by including context as a new feature of a data set. In figure 3.1, which represents a simple example, point *X1* and *X2* are global anomalies, *X3* is a local anomaly and *C3* expresses a small anomalous cluster. Goldstein [3] specifically describes that based on the example in figure 3.1, algorithms that output a score are much more useful than binary labeling algorithms.

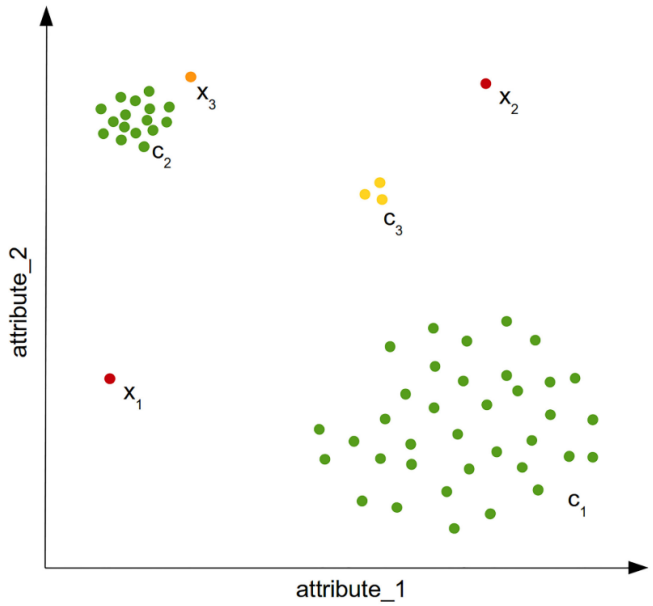

Figure 3.1: Example of outlier types, distinguished between global outliers *(x1, x2)* and local outliers *(x3)* [3]

3.3.2 Labeling and Scoring

Generally outlier detection algorithms can produce two types of output for a specific data point which can be a binary label or a real-valued score [3], [8], [10], [29]. A binary value output would be an indication whether a data point is an outlier or not, where a real-valued outlier score could represent an absolute score or probability score. Scores are more common, especially in an unsupervised setting, due to the fact that it is more practical to provide the top anomalies to the user [3]. Furthermore a score usually provides more information indicating a degree of abnormality, also scores can be converted to a label by the use of a

threshold.

3.3.3 Proximity-based

Examples of proximity-based algorithms are KNN [30], Local Outlier Factor (LOF) [31], Local Outlier Probability (LoOP) [32], Local Correlation Integral (LOCI) [33] and K-modes [34]. KNN is mostly suitable for global anomalies and computes the average distance to the k nearest neighbors [35] (multiple points) or k'th nearest neighbor (single point) of a data point [3], [30]. LOF is suitable for local outliers where a ratio of local densities is computed [3], [31]. LOF is local since it only relies on its direct neighbors and the outlier score is based on the k neighbors only. LoOP is similar to LOF but computes a probability score for each data point, since with LOF one has to set a threshold to decide whether a data point is anomalous or not. Choosing k is crucial in both KNN and LOF therefore LOCI utilizes all possible values of k in order to compute an anomaly score [3], [33]. K-modes [34] is clustering technique specifically designed for categorical data. Suri [36] adjusted the algorithm in order to be able to detect outliers based on the distance to the centre of the clusters.

3.3.4 Subspace Techniques

An example of a subspace techniques is PCA [37]. With PCA one computes the eigenvectors and eigenvalues of a numerical data set in order to be able to transform the data to one of it's subspaces [3], [10]. Observations in the data are aligned along an eigenvector and data points that do not respect this alignment can be assumed as outliers [10]. Outlier scores can be obtained as the sum of the projected distance of a sample on all eigenvectors [21].

3.3.5 Statistical / Probabilistic Models

Statistical / probabilistic models can be Mixture Modeling and HBOS [3], [10]. The principle of mixture modeling, using the expectation-maximization algorithm, is to assume that the data set was generated from a mixture of k distributions [10]. Parameters of multiple generative models, usually Gaussian distributions, are estimated so that the observed data has a maximum likelihood fit [10]. These models can be used to estimate probabilities of the underlying data points, anomalies will have very low fit probabilities. HBOS [22] is a statistical algorithm assuming independence of features and computes histograms of each feature. Based on the heights of the bins a outlier score can be computed by multiplying the inverse of bin heights a data point resides in [3].

3.3.6 Other

Other selected algorithms are: IF [20], One-class SVM [38] and Autoencoder Neural Networks [2]. An IF is an ensemble of a set of isolation trees. In an isolation tree [20], the data is recursively partitioned with axis-parallel cuts at randomly chosen partition points in randomly selected attributes, so as to isolate the data point into nodes with fewer and fewer

data points until the points are isolated into singleton nodes containing one data point [10]. In such cases, the tree branches containing outliers are noticeably less deep, because these data points are located in sparse regions. Therefore, the distance of the leaf to the root is used as the outlier score [10]. The technique shares some intuitive with another ensemble technique known as random forests, which is very successful in classification problems [10].

One-class SVM [38] attempt to learn a decision boundary that achieves the maximum separation between data points and the origin this results in some kind of hulls describing the normal data in the feature space [3]. A data point is scored anomalous if it's distance to the determined decision boundary is high.

Finally an Autoencoder Neural Network can be used in order to detect anomalies in a data set. Autoencoder Neural Networks are feed-forward multilayer networks that can be trained to reconstruct its input [2]. The difference between the reconstructed output and the original input can be used to compute an outlier score this is referred to as reconstruction error [2]. The intuition behind this is that an Autoencoder learns the underlying main aspects of a data set and therefore outliers will have a high reconstruction error.

Category	Data type	Name	Output	Paper	Technical paper and no. citations
Proximity-based	Numerical	K-Nearest-Neighbor	Score	[3], [10]	
				[26], [39]	
				[21], [40]	
				[30], [35]	1918 [30]
		Local Outlier Factor	Score	[3], [10]	
				[26], [40]	
				[21]	4095 [31]
		Local Outlier Probability	Probability	[3], [32]	206 [32]
		Local Correlation Integral	Score	[3], [10]	
				[21], [40]	909 [33]
	Categorical	K-Modes	Score	[36], [41]	
				[42]	598 [34]
Subspace techniques	Numerical	Principal Component Analysis	Score	[3], [10]	
				[39], [40]	
				[21]	499 [37]
Statistical / probabilistic model	Numerical & categorical	Mixture Modeling	Probability	[10], [43]	
				[26], [39]	
				[40]	2318 [44]
		Histogram-based Outlier Score	Score	[3], [10]	
				[21]	58 [22]
Other	Numerical	Isolation forests	Score	[10], [21], [45]	507 [20]
		One-class Support Vector Machine	Score	[3], [10]	
				[26], [39]	
				[21], [40]	207 [38]
		Autoencoder Neural Network	Score	[2], [10]	
				[26], [39]	
				[21], [40]	11 [2]

Table 3.1: Overview of unsupervised outlier detection techniques

3.4 Discussion

Combining multiple studies analyzed during this literature study it became evident that a wide variety of unsupervised outlier detection techniques exists. During this research an attempt has been made to list unsupervised outlier detection techniques which are suitable specific for transactional audit data. Goldstein and Uchida evaluate 19 different unsupervised outlier detection techniques on 10 data sets from different applications domains [3]. Some of the techniques considered during their research are extended or different versions of the same technique, therefore some of these techniques aren't listed in this research. Also in a book about outlier analysis by Aggarwal describes outlier detection techniques [10] that haven't been included in this literature study. A noteworthy technique worth mentioning is the use of outlier ensembles [10]. Ensemble techniques are popular methods used to improve the accuracy of various data mining techniques. These techniques combine the outputs of multiple algorithms in order to generate a unified output. The idea is that some techniques will do better on a particular subset of the data whereas other techniques will do better on other subsets of the data [10].

During this literature research only Autoencoder Neural Networks have found to be extensively experimented with in an equal problem setting, being the detection of outliers in transactional audit data consisting of journal entries. Schreyer et al. compared the results of this technique with four other algorithms and the results show that Autoencoder Neural Networks is superior in terms of outlier detection performance [2]. Schreyer et al. successfully experimented with only two data sets therefore this technique has enough room to experiment even further with. The fact that, based on this literature study, only a single unsupervised outlier detection technique has been studied and tested on transactional audit data in order to aid auditors during financial audit processes, indicate that there is a lot of room for further research. Confirming the statement of Appelbaum et al. in 2018 that more research could be applied on the application of data mining techniques in many of the audit phases [6].

3.5 Conclusion

The goal of this literature study has been to explore the field of unsupervised outlier detection techniques that could be applied during financial audits. Outlier detection could be of use during audit processes potentially enabling the auditor to execute audits more effectively and more efficiently by providing them with a set of outliers, being observations that appear to be different from the norm.

In this section multiple unsupervised outlier detection techniques that could find usage during financial audits have been described and listed. These techniques are categorized in proximity-based, subspace, statistics/probabilistic and other models. A systematic literature study has been conducted in order to come up with an overview of well known techniques. From all the studied papers only one focused on the application of unsupervised outlier detection during financial statement audits with transactional data consisting of journal en-

tries. Furthermore it has been found that techniques such as Neural Networks, clustering and decision trees require more research in the field of external audits. Therefore it can be concluded that many research has been done in the field of unsupervised outlier detection but not much research has been done applying these techniques to financial audit data. The previously described techniques have proven to be successful on a various data but this can not be said about applying these techniques to transactional data and specifically financial audit data. This research provides an overview with unsupervised outlier detection techniques that require further research and experimenting in order to find whether these techniques find there value during financial audit processes.

Chapter 4

Research Methodology

The following sub research questions have been formulated for this research:

RQ1 *Which (class of) unsupervised outlier detection techniques can be applied on transactional audit data in order to detect outliers?*

RQ2 *Which unsupervised outlier detection technique performs best in detecting outliers that are of interest for the auditor?*

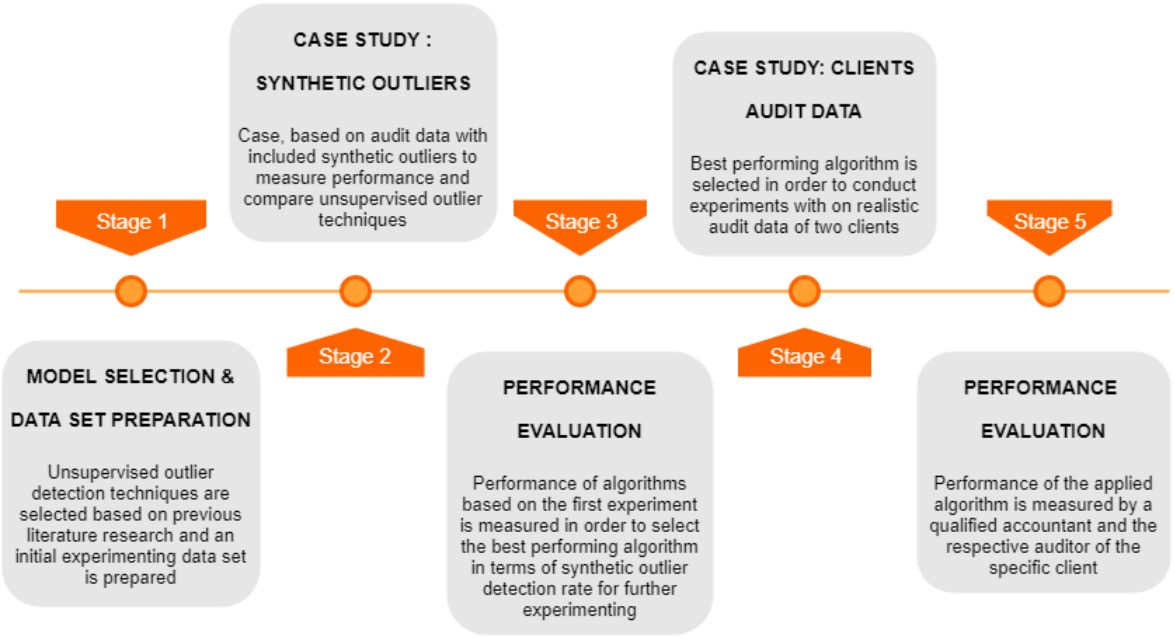

Figure 4.1: Research stages

To answer the described research questions an exploratory research will be utilized. An exploratory research is described as the process of gathering background knowledge in order to expand the understanding of the research field and discover future research tasks [46]. Furthermore with an exploratory research useful results are hoped for but are not guaranteed [47]. Experiments, cases, with multiple unsupervised outlier detection algorithms will

be conducted in order to discover the characteristics and possible usability of the algorithms during financial statement audits. An overview of the stages in the research process are visualized in figure 4.1.

4.1 Stage 1: Model selection & Data Set Preparation

Unsupervised outlier detection techniques that could be applied to detect outliers in an unsupervised manner are presented in section 3.3, table 3.1. These techniques are listed based on a systematic literature review (described in section 3.1). From the 11 algorithms presented as a result of the literature study, a selection of four algorithms will be made that are subject to experiments. This selection is made for the reason that not all algorithms can be experimented with in the given the amount of time.

Procedure of selection is based on the category an unsupervised outlier detection technique resides in. Therefore, to answer the first sub research question **RQ1**, from each category an outlier detection algorithm will be selected based on results presented previous academic literature. To select unsupervised outlier detection techniques from the listed categories, a set of criteria is determined. To be included in initial experiments a model in each category is selected based on the following criteria:

C1 *Successfully applied in recent research ($\geq$ 2014) to detect outliers in transactional audit data*

C2 *Successfully applied in recent research ($\geq$ 2014) on data sets containing categorical features*

C3 *Favorable results in recent research papers ($\geq$ 2014)*

First, only recent comparative studies are taken into account in order to perform a proper and clear selection of unsupervised outlier detection techniques. The definition of a recent study in the context of this thesis would be papers from 2014 or newer. It is believed that recent comparative studies provide more reliable insights when comparing machine learning algorithms in particular. For example, a machine learning model that performed very well 10 years ago might now be obsolete. It is a field that develops very quickly, new improved techniques arrive at a high rate. Therefore only recent comparative studies are taken into account in order to select proper outlier detection techniques based on recent result.

Unsupervised outlier techniques that have been applied in an equal problem setting and are proven to be successful in recent research are selected at first (**C1**). The definition of a similar problem, would be the application of an unsupervised outlier detection technique on transactional audit data consisting of journal entries. The second criterion has been the selection based on evaluating results where unsupervised outlier techniques are applied on data sets containing categorical features (**C2**). This because transactional audit data is particularly facet of categorical data and outlier detection techniques that perform well on this type of data, might perform well on transactional audit data. The last criterion, if no model of

a category meets the previous criteria a selection is made based on favorable results (**C3**), being that comparative results generally prefer a given model.

4.1.1 Data Preprocessing

The selected four unsupervised outlier techniques are subjected to a series of single-case mechanism experiments [47]. The initial experiments are based on a transactional audit data set in which synthetic outliers have been injected in order to be able to evaluate the outlier detection technique in terms of synthetic outliers detected. General process of preprocessing is visualized in figure 4.2.

In preparation of the first experiments, a transactional audit data set has been selected. This data set is contains transactional audit data from one of the smaller clients of de Jong & Laan and is selected by a CPA, a qualified accountant from de Jong & Laan. This transactional data set is a relatively small but of realistic size and consisting of $4,879$ journal entries, each journal entry contains a varying amount of mutations, with a total data set size of $12,882$ mutations. This size is chosen in order to reduce the execution time of algorithms and therefore ensure a more efficient method of experimenting. The average amount of mutations according to a CPA of de Jong & Laan lies between $100,000$ and $250,000$.

The initial data set will contain injected synthetic outliers, this technique is applied during similar research by Schreyer [2] and allows to measure performance in terms of detected outliers. These outliers are injected by the CPA where distinction is made between global and local outliers, the same as Schreyer did during his research. Together with the CPA a small amount of 7 (0.14 %) anomalous journal entries has been inserted in the data set. The injected anomalous transactions consist out of 18 mutations in total. The distinction between local and global outliers is made by the CPA. Besides, inserted outliers are all journal entries that, according to the CPA, are of interest during a financial statement audit processes. 2 out of 7 outlying journal entries are considered local anomalies and 5 out of 7 are considered global anomalies.

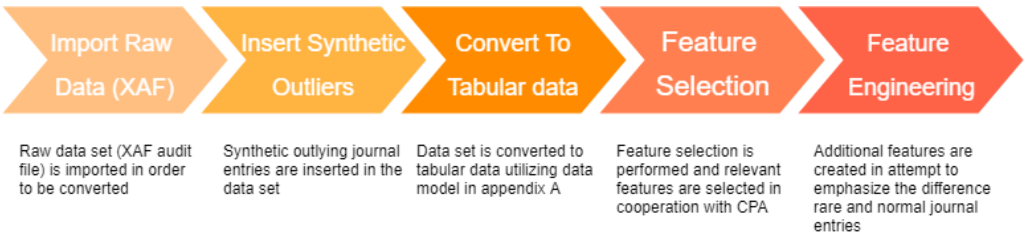

Figure 4.2: Steps data preprocessing initial experimenting data set

As transactional audit data is extracted from the client's AIS it is in standardized XAF format (see background section 2.1.2). The synthetic outliers are injected and utilizing the data model (appendix A), the audit data can be represented in tabular structure.

The data set contains multiple features from which irrelevant features are removed. This results with 6 relevant features from the data set, further elaborated in the following chapter. These features are: *transaction amount, transaction date, transaction type, cus-*

tomer/supplier ID, journal ID, general ledger ID. The process of feature selection is performed with the assistance of two CPAs. The features that have proven to be the most relevant during financial statement audits have been selected based on both the CPAs experience.

Following feature selection, additional 15 features have been created based on the selected features, these are explained in detail in the following chapter. Some of the created features are elaborated in even more detail in Appendix C. During this research, feature engineering has been a continuous process and new features have been created during the experiments with unsupervised outlier detection techniques. The additional features have been engineered in collaboration with two CPAs. The goal has been to create features that emphasize the difference between a rare instance and a normal instance. During experiments, different combinations of features have been experimented with in order to find the better combination of features. With a total of 21 different feature experiments are being conducted with 11 different combinations of features, being different sub-selections of the 21 features. These feature combinations are presented in table 6.2 in chapter 6.

4.2 Stage 2: Case Study: Synthetic Outliers

Experiments with four unsupervised outlier detection techniques are conducted on the data set containing synthetic outliers. Different feature combinations are tested with in order to improve detection performance. The techniques experimented with are: Isolation Forests (IF), K-Nearest Neighbor (KNN), Histogram-based Outlier Score (HBOS) and Autoencoder Neural Networks. The goal of the experiments has been to be able to compare the techniques based on the performance of synthetic outliers detected, from the previous described data set.

For some of the outlier detection methods the categorical variables of the data set required a numerical encoding and/or normalization. Encoding methods for each technique are as following:

IF *Labelencoding, encode categorical variables to a value 0 and n unique variables*

KNN *Labelencoding, encode categorical variables to a value 0 and n unique variables. Data set is normalized using min-max normalization*

HBOS *Data set is not encoded nor normalized*

Autoencoder Neural Network *One-hot encoding, generates new (binary) columns, indicating the presence of each possible value from the original data with 0 or 1. Data set is normalized using min-max normalization*

Alongside different feature combinations different hyperparameter settings are used during experiments. Table 4.1 presents an overview of the outlier detection techniques along with the relevant hyperparameters from which some have been experimented with.

	Language / Tool	Hyperparameters
Isolation Forests	*Python*	*max_features = 0.75 / 1.0,* *n_estimators = 100 / 250 / 500,* *max_samples= 1.0 ,* *bootstrap = False*
K-Nearest Neighbor	*Python*	*n_neighbors = 10 / 25 / 50 / 100 / 150,* *method = mean*
Histogram-Based Outlier Score	*RapidMiner Studio*	*binwidth = static,* *n_bins = -1*
Autoencoder Neural Network	*Python*	*Leaky Rectified Linear Units (LReLU); a = 0.4,* *Batch-size = 128,* *Optimizer = Adam,* *Learning rate = 0.0001,* *Weight initialization = Glorot Normal*

Table 4.1: Implementations and hyperparameters selected for evaluation

In case of IF the tune-able parameter has been the percentage of features the algorithm samples *max_features* from for each base estimator which has either been 75% or 100%. The number of base estimators *n_estimators*, being the number of constructed trees for each individual data entry has been set to 100, 250 or 500. *Bootstrap* is set to false, therefore the algorithm trains on each individual data entry rather than sampling from the entire set with replacement. Other parameters are kept at a default setting.

In case of KNN a tune-able parameter has been the number of **K** neighbors which have been evaluated at either 10, 50, 100, 150 or 200. Initially the rule of thumb, being that **K** should be $10 < k < 50$ [3], was followed but also expanded utilizing a **K** of 100 and 150. *Method* has been set to 'mean', which means the average of all **K** neighbors determines the outlier score. Other parameters are kept at a default setting.

In case of HBOS, the number of bins **n_bins** has been set to *-1*, resulting that the number of bins is calculated by $\sqrt{n}$, n being the number of data points in the set [22]. For categorical features simple counting of the values of each category is performed and the relative frequency is computed [22].

The Autoencoder Neural Network has been experimented with most extensively due to the fact of proven success in previous literature. Varying hyperparameters and architectures which are elaborated in further detail in appendix E. Different architectures of layers and neurons have been experimented with based on previous literature. Evaluated architecture ranged from shallow architectures to deep architectures and bottleneck sizes ranging from 3 up to 40 neurons. The loss functions that have been used during experiments is Cross-entropy [2], Mean Squared Error (MSE) and a combination of these two. The activation function of the hidden layers has been set to LReLU with a scaling factor of $\alpha = 0.4$, optimizer used has been the Adam optimizer with a learning rate equal to $\eta = 10^{-4}$ [2]. Weight initialization used is Xavier Initialization, which is also the same as Schreyer did during his research [2]. Multiple experiments have been conducted with the Autoencoder Neural Network using different architectures, number of epochs, and loss functions.

4.3 Stage 3: Performance Evaluation

In order to evaluate and compare the previous described unsupervised outlier techniques, multiple steps are taken. Initially, all techniques evaluated produce a score as output for each individual mutation in the audit data set (outlier score). The scores are normalized to a value between 0 and 1 indicating a degree of abnormality where 0 is normal and 1 is abnormal.

Next the results are aggregated by only picking the maximum scoring mutation of each journal entry. Meaning that results are evaluated on journal entry level rather than the level of a single mutation. An example of a scored journal entry is given in table 4.2.

tran_Nr	rgl_trDt	rgl_amt	rgl_amntTp	cust_SupId	cust_SupName	dgb_jrnId	gb_accId	Outlier Score
21515	2019-08-03	567.32	C			2200	2200	0.521
21515	2019-08-03	526.08	D	cXXXX	Supplier X	2200	1500	0,474
21515	2019-08-03	41.24	D	cXXXX	Supplier X	2200	1500	0.503
21533	2019-08-07	321.26	D			2205	2205	0.846
21533	2019-08-07	331.2	C	dYYYY	Customer Y	2205	1300	0.957
21533	2019-08-07	9.94	D			2205	2205	0.456

Table 4.2: Example of two journal entries with an outlier score for each mutation

After aggregation the result is as following (table 4.3), so only the top scoring mutations are left as a result. This is done for the fact that if a single mutation within a journal entry has a high score, an auditor will always analyze the entire journal entry.

tran_Nr	rgl_trDt	rgl_amt	rgl_amntTp	cust_SupId	cust_SupName	dgb_jrnId	gb_accId	Outlier Score
21515	2019-08-03	567.32	C			2200	2200	0.521
21533	2019-08-07	331.2	C	dYYYY	Customer Y	2205	1300	0.957

Table 4.3: Result after aggregation, respective highest scoring mutation of journal entries are selected and considered in evaluation process

During experiments in this research only the synthetic injected outlying journal entries are known. The results of the experiments are therefore only analysed and evaluated in terms of synthetic outliers detected. Detection performance is analyzed on how many of the highest scoring journal entries have to be selected in order to obtain a recall percentage of 100%, indicating that all actual positives (synthetic outliers) have been identified.

4.4 Stage 4: Case Study: de Jong & Laan Client Audit Data

From previous experiments, the best performing unsupervised outlier detection technique is selected for further experimenting. Two clients of de Jong & Laan have been selected, guided by one of the CPAs of de Jong & Laan. These experiments on both audit data sets have been conducted during a live audit process of these clients and results have been evaluated shortly after finishing the audit process. This ensures that the CPA auditors of these clients still had specific knowledge relevant to the client's audit process in mind. The

setting of the experiments has been fully unsupervised, meaning that no labeled data has been available.

In comparison to the previous studied case, the two client audit data sets are significantly larger. One audit data set contains $15,376$ individual transactions and $342,820$ records. The other data set containing $8,744$ individual transactions and $40,004$ records. Data sets come from two different types of clients, both coming from different business sectors.

4.4.1 Experiments

During the previous case IF has been selected as the best performing outlier detection technique based on the detection rate of the synthetic outliers. The percentage of features (*max_features*) the algorithm samples from for each base estimator which has been set at 75%. The number of base estimators *n_estimators* has been set to 100. *Bootstrap* is set to false other parameters are kept at a default setting. The combination of features that has been selected during both experiments and as a result of previous experiments are presented in table 4.4. Again, the categorical features required encoding, this is done by using a labelencoder.

Feature Name	Description	Data type
cf_dateIndex	Number of days passed since the first transaction in the audit data set	Numerical
cf_debCredAmntLog	The log transformed amount of the transaction, being either debit (positive) or credit (negative)	Numerical
cf_custMean	The mean amount for a specific relationship a transaction applies to	Numerical
cf_distanceToMean	The distance to the the mean of the specific relationship	Numerical
cas_DescDet3	General ledger the specific mutation relates to	Categorical
cust_SupId	Relationship the specific mutation relates to	Categorical

Table 4.4: Selected features during both client experiments

4.5 Stage 5: Performance Evaluation

To evaluate the results of IF applied on client audit data thee experts from de Jong & Laan have been consulted. The auditor of each client evaluated the results of the unsupervised outlier detection technique along with a CPA who analysed the results of both clients.

Outlier scores are again normalized to a value between 0 and 1 indicating a degree of abnormality. A threshold for both client data sets is set at 0.6, journal entries having mutations with a higher outlier score than the threshold are considered anomalous.

The outliers of the client data sets are presented to the associated auditors. The next step has been for the auditors to determine whether individual transactions could be of interest during financial statement audits. The auditor has very specific client knowledge and specific knowledge about the audit data set which allows him to analyse the results in a very detailed way. Each outlying journal entry has been analyzed manually and it is determined whether a transaction could be of interest or not. A transaction could be of interest for an auditor if the auditor finds the journal entry suspicious enough willing to perform extra research based on the journal entry alone. The reason of this could be that the auditor

suspects that administrative errors have been made or the transaction is suspicious in terms of fraud.

As a result of the previous evaluation process, an absolute list of detected anomalous journal entries is labeled with being of interest or not (considered anomalous by both CPA and auditor). Alongside the auditor indicated whether the transaction, detected by the outlier detection technique, has all ready been detected during the regular audit process.

Finally, a small research has been done to the specific journal entries that have been investigated based on regular audit procedures. The auditors of both clients investigated a selection of journal entries based on statistical methods. These have been compared with the results of the IF algorithm to find their corresponding outlier scores.

Chapter 5

Model selection and Data Set Preparation

In this chapter a selection of four unsupervised outlier detection techniques is made. In order to evaluate the potential of an outlier detection technique a transactional audit data set is prepared containing synthetic outliers to enable performance measurement in terms of detected outliers. From this transactional data set, features have been selected and additional features have been generated.

5.1 Unsupervised Outlier Detection Model Selection

From the listed unsupervised outlier detection categories in table 3.1 in chapter 3, being: *proximity-based techniques, subspace techniques, statistical / probabilistic techniques* and three techniques categorized as *'other'*, a selection of four algorithms has been made. From each category an technique is selected, the selection procedure is generally visualized in figure 5.1. In the figure visualizes the selection procedure within each category according to multiple criteria defined in the previous chapter.

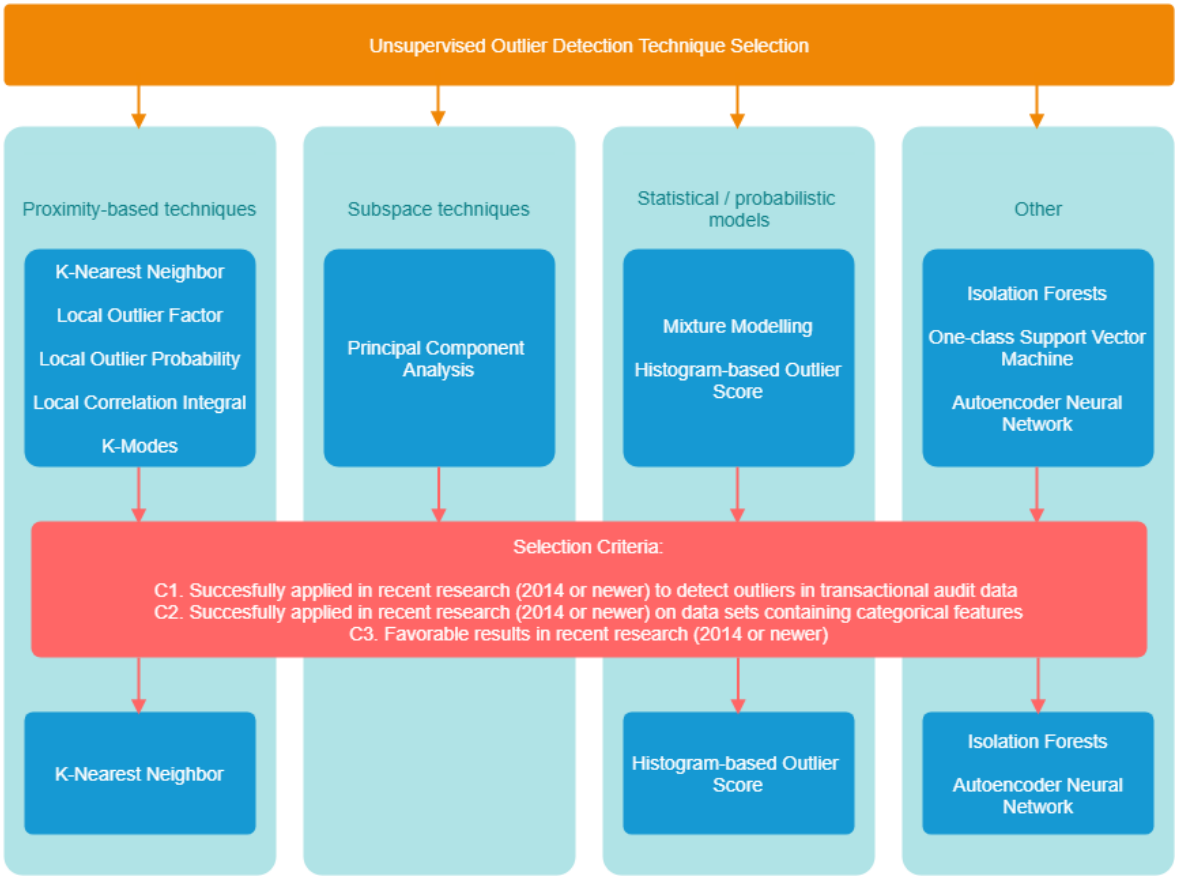

Figure 5.1: Selection process of unsupervised outlier detection techniques

Within each category a technique has been selected meeting at least one of the criteria. An exception has been made for the category 'subspace techniques' only containing PCA, due to consistently under-performing compared to other categories [2], [3], [48]. In a research of Lazarevic et al. [48], LOF, KNN and PCA and one-class SVM have been compared in a supervised network intrusion detection setting and PCA had the lowest detection rate compared to the other techniques. Goldstein and Uchida [3] compared a 19 different unsupervised outlier detection techniques on 10 different data sets. The results show that PCA, on 9 of the 10 data sets, performed worse than other unsupervised outlier techniques. Finaly, Schreyer et al. [2] applied Autoencoder Neural Networks to detect outliers in transactional audit data, being an equal problem setting as presented in this thesis, and compared the results with 4 other unsupervised outlier detection techniques one of them being PCA. PCA had the poorest detection performance compared the other unsupervised outlier detection techniques. Given the exploratory nature of this study it has been decided to drop the category 'subspace techniques' based on overall underwhelming performance in unsupervised outlier detection problems.

Based on the comparative study of Goldstein and Uchida [3] KNN has been the unsupervised outlier detection technique of choice for the proximity-based category. During this study KNN seems to be slightly favourable over LOF in detecting outliers on data sets where the percentage of outliers is lowest. These data sets have an outlier percentage of 0.17%

and 0.15%, therefore having a similar outlier percentage as the first single case experiments. Among the proximity based methods the study by Goldstein and Uchida concludes that KNN is the best performing unsupervised outlier detection technique on average.

Similarly based on research of Goldstein and Uchida [3] HBOS has been the algorithm of choice in case of statistical / probabilistic models. Based on the results of reported in the research of Goldstein and Uchida HBOS has been the best outlier detection technique of all included outlier detection techniques in the study. HBOS has been the best performing algorithm on four of the 10 data sets which is better than any of the other compared algorithms. Besides HBOS does not require any encoding of categorical features and is therefore more suitable in case of transactional audit data. HBOS therefore meets both the second criterion and the third criterion.

Alongside the previous categories three unsupervised outlier detection techniques do not belong to a specific category and are categorized as 'other'. From these techniques Autoencoder Neural Network and IF has been selected for inclusion in the single case experiments. Autoencoder Neural Network has been applied for the detection of outliers in an unsupervised setting during a very recent research by Schreyer et al. [2]. During this research the technique has been applied in the exact same setting as presented in this thesis, namely the detection of anomalies in transactional audit data. The technique has been applied on two data sets also containing journal entries, very similar to the data sets used in this thesis. Schreyer et al. successfully experimented with Autoencoder Neural Networks by injecting synthetic outliers into both data sets in order to make the problem supervised. The results reported show that an Autoencoder Neural Network has a better detection rate than other unsupervised outlier detection algorithms studied in the same research. Based on the results of Schreyer et al. and the application of the technique on a similar problem Autoencoder Neural Network is also included in these single case experiments.

IF has been included due to a very recent comparative evaluation of Domingues et al. [45]. Domingues et al. conclude that IF is an excellent method to efficiently identify outliers while having an excellent scalability on large data sets. Due to being a well performing unsupervised outlier detection technique and the possibility easily scale the algorithm to larger data sets this technique is also included in the first single case experiments.

5.2 Data Set Preparation

Based on the previous section, four unsupervised outlier detection techniques have been selected in order to conduct experiments with. Following this selection a single initial transactional audit data set is prepared for experimenting. The transactional audit data set comes from one of the audit clients of de Jong & Laan Accountants and is therefore a realistic data set containing real transactions. The data has been extracted from the client's AIS and is in standardized format, being XAF format. Following the database diagram from appendix A, tabular data is created from which a sample and relevant features are presented in appendix B. The transactional data set contains $4,879$ journal entries with a total of $12,882$ mutations

(synthetic outliers included).

5.2.1 Feature Selection and Feature Engineering

In preparation of experiments, irrelevant features are removed and additional features are created. Table 5.1 represent the features selected from the original audit data set (XAF audit file).

Feature Name	Original Name (XAF)	Description	Data Type
rgl_Amt	Amount	Mutation amount in local currency	Numerical
rgl_trDt	Transaction Date	Date the transaction was posted in the AIS of the audit client	Date
rgl_amntTp	Transaction Amount type	Indication whether the amount is either debit or credit	Binary
cust_SupId	Customer Supplier ID	Identification of a customer or supplier to which a transaction applies to. Contains ”internal” in case of no customer or supplier	Categorical
dgb_JrnId	Journal ID	Identification of a daybook to which the transaction belongs	Categorical
cas_DescDet3	CaseWare Description	Description of standardized general ledger names	Categorical

Table 5.1: Selected features, including description, data type and nullable type

Table 5.2 represent all engineered features. Some of the engineered features are explained in more detail in appendix C. Again, feature engineering has been a continuous process during these single-case experiments comparing four unsupervised outlier detection techniques.

Custom Feature Name	Description	Data Type
cf_Year	The number of the year in which the transaction took place	Numerical
cf_Month	The number of the month in which the transaction took place	Numerical
cf_Day	The day of the month at which the transaction took place	Numerical
cf_Weekday	The day of the week at which the transaction took place	Numerical
cf_IsWeekend	Whether a transaction took place in a weekend or not	Binary
cf_dateIndex	Number of days passed since the first transaction in the set	Numerical
cf_Day_sin / cf_Day_cos	Sine and cosine transformation of cf_Day	Numerical
cf_Weekday_sin / cf_Weekday_cos	Sine and cosine transformation of cf_Weekday	Numerical
cf_Month_sin / cf_Month_cos	Sine and cosine transformation of cf_Month	Numerical
cf_dateIndex_sin / cf_dateIndex_cos	Sine and cosine transformation of cf_dateIndex	Numerical
cf_custMean	Mean of the amount from all transactions of a specific customer or supplier	Numerical
cf_distanceToMean	Difference of the amount of a transaction to cf_custMean	Numerical
cf_tranSum	The sum of each mutation in a transaction	Numerical
cf_debCredAmnt	In case transaction is of type 'credit' the amount is transformed to a negative number else it is just the amount	Numerical
cf_debCredAmntLog	Log transformation of cf_debCredAmnt	Numerical

Table 5.2: Additional engineered features

5.2.2 Synthetic Injected Outliers

The synthetic anomalous journal entries injected in the initial audit data set are shortly listed in this section. The tables presented in appendix F present injected synthetic anomalous journal entries in more detail. The outlier type is indicated alongside a description of the properties of the journal entries. A CPA inserted the outlying journal entries each with a different intention therefore making the outliers unique and either globally or locally anomalous. According to the CPA, the outliers are typical examples of transactions that could be of interest during a financial audit, meaning that from an audit perspective the transactions have a

high potential to be subjected to further research. Referring to section 3.3.1 data points that are very different from dense areas with respect to their attributes are called global anomalies, hence a data point is only anomalous when compared to its close-by neighbors it is called a local anomaly.

Local; increase of debt to supplier *In relation to a sub-selection of the entire transactional data set this synthetic outlier has a transaction amount relatively large in comparison to average of all transactions from the same supplier. Also the amount of the transaction is relatively large in comparison to the average of transactions posted on the specific ledger account*

Local; delivery of products *In relation to a sub-selection of the entire transactional data set this synthetic outlier has an amount that relatively large in comparison to average of all transactions from the same customer. Also the amount of the transaction is relatively large in comparison to the average of transactions posted on the specific ledger account*

Global; payment to a supplier *In relation to the transactional data set this synthetic outlier has a transaction date after the financial year. Generally a financial statement audit is executed on a data set from a single previous financial year, transactions with a transaction date in a different year would be considered very anomalous*

Global; payment from a supplier *In relation to the transactional data set this synthetic outlier has a transaction date after the financial year. Generally a financial statement audit is executed on a data set from a previous financial year, transactions with a transaction date in a different year would be considered very anomalous*

Global; payment to a supplier *In relation to the transactional data set this synthetic outlier has a transaction date before the financial year. Generally a financial statement audit is executed on a data set from a previous financial year, transactions with a transaction date in a different year would be considered very anomalous*

Global; payment to supplier *In relation to the transactional data set this synthetic outlier has an amount that is one of the extreme large values in the entire set*

Global; payment from customer *In relation to the transactional data set this synthetic outlier has an amount that is one of the extreme large values in the entire set. Alongside, the journal entry is not in balance which would considered very rare. The total debit amount should be equal to the total credit amount in a journal entry*

Chapter 6

Results Case Study: Synthetic Outliers

Following the experiments conducted, results are presented in this chapter. The evaluation process will be briefly explained in the following section followed by the results.

6.1 Single Case Experiments

Following model selection and the preparation of the experimenting data set, experiments have been conducted. Table 6.1 presents the properties of the data set containing synthetic outliers along with the percentage of synthetic outliers. The percentage of synthetic outliers are 0.14%. The following chapter presents the results of experiments and in order to evaluate the performance of the techniques. The techniques have been evaluated in terms of outliers detected given a recall percentage of 100% for the synthetic outliers.

	Journal Entries	**Transactions**	**Percentage**
Unlabeled data	4,872	12,864	99.86 %
Global synthetic outliers	5	13	0.1 %
Local synthetic outliers	2	5	0.04 %
Total synthetic outliers	7	18	0.14 %
Total	4,879	12,882	100 %

Table 6.1: Properties of the synthetic data set

The table 6.2 represents all feature combinations that have been experimented with. The feature combinations are numbered and referred to as **'FC1'** up untill **'FC11'**.

		Feature Combination										
		FC1	FC2	FC3	FC4	FC5	FC6	FC7	FC8	FC9	FC10	FC11
rgl_amntTp	Indication whether the amount is either debit or credit			X	X	X	X	X				
cust_SupId	Identification of a customer or supplier to which a transaction applies to					X	X	X	X	X	X	X
dgb_jrnId	Identification of a daybook to which the transaction belongs					X	X					
cas_DescDet3	Description of standardized general ledger names					X	X			X	X	X
rgl_amt	Mutation amount in local currency	X	X						X	X		
cf_Year	The number of the year in which the transaction took place		X	X		X			X	X		
cf_Month	The number of the month in which the transaction took place		X	X		X						
cf_Day	The day of the month at which the transaction took place		X	X		X						
cf_Weekday	The day of the week at which the transaction took place		X	X		X						
cf_IsWeekend	Whether a transaction took place in a weekend or not		X	X		X						
cf_dateIndex	Number of days passed since the first transaction in the set		X	X		X			X	X	X	X
cf_Day_sin \cf_Day_cos	Sine and cosine transformation of cf_Day	X		X	X		X	X				
cf_Weekday_sin \cf_Weekday_cos	Sine and cosine transformation of cf_Weekday	X		X	X		X	X				
cf_Month_sin \cf_Month_cos	Sine and cosine transformation of cf_Month	X		X	X		X	X				
cf_dateIndex_sin \cf_dateIndex_cos	Sine and cosine transformation of cf_dateIndex	X		X	X		X	X				
cf_debCredAmnt	In case transaction is of type 'credit' the amount is transformed to a negative number else it is just the amount			X	X	X	X	X			X	
cf_custMean	Mean of the amount from all transactions of a specific customer or supplier									X	X	X
cf_distanceToMean	Distance of the amount of a transaction to cf_custMean							X		X	X	X
cf_tranSum	The sum of each mutation in a transaction									X	X	
cf_debCredAmntLog	Log transformation of cf_debCredAmnt											X

Table 6.2: Combinations of features used for experiments

6.2 Evaluation

In order to evaluate the performance in terms of detected synthetic outliers the data set is aggregated by picking only the maximum outlier score of each journal entry's mutations. After aggregation the data set has a size of $4,879$, being the highest scoring mutations within each journal entry.

In order to analyse the detection performance a threshold is determined in order to obtain a recall percentage of 100% for all injected synthetic outliers. In other words, the threshold is to be set equal to the value of the lowest scoring synthetic journal entry. Every journal entry scoring equal or higher than this score is to be considered 'outlier'. Results are presented in the amount of outliers detected given the threshold, where a lower amount of outliers detected would be preferred. This procedure has been executed for all 7 synthetic outliers and individually for the 5 global synthetic outliers and for the 2 local synthetic outliers.

6.3 Results

Table 6.3 presents an overview of all unsupervised outlier detection methods experimented with on the transactional data set containing synthetic outliers. The full results of all techniques with all feature combinations and hyperparameters are presented in appendix D. The top number of outlying journal entries for each feature combination and corresponding best performing hyperparameters are presented. Table 6.3 presents all outlying journal entries, given a threshold to retrieve a recall of 100% for all 7 injected synthetic outlying journal entries. The percentage of outliers describe the percentage of outlying journal entries with respect to the entire data set of $4,879$ journal entries.

Based on the results it can be found that Isolation Forests (IF) generally outperforms other unsupervised outlier detection techniques on average. IF is only being outperformed by HBOS and / or KNN on feature combination **FC2, FC3, FC5**. Given IF and feature combination **FC8**, only the top 23 journal entries have to be selected in order to obtain a recall percentage of 100% and include all 7 synthetic outliers. Figure 6.1 represents a graphical view of the percentage of outliers detected in the data set for each outlier detection technique. The figure clearly shows that HBOS and IF perform very consistent over all feature combinations. KNN seems to underperform on feature combinations **FC5, FC6, FC7, FC10**. Inspecting these feature combinations (see table 6.2) it can be found that these contain more categorical data in comparison to the other feature combinations. Overall Autoencoder Neural Network performs worse than any of the other compared unsupervised outlier detection techniques. The Autoencoder Neural Network comes close to the other techniques on feature combination **FC2** and is better than KNN on **FC7**.

Table 6.4 present the top number of outlying journal entries in order to obtain a recall of 100% only for the global synthetic outliers. Table 6.5 presents results in the same structure but only for the local synthetic outliers. These tables will be further elaborated below.

	HBOS [No. Outliers]	HBOS [% Outliers]	Autoencoder [No. Outliers]	Autoencoder [% Outliers]	KNN [No. Outliers]	KNN [% Outliers]	IF [No. Outliers]	IF [% Outliers]
FC1	284	5.82	1053	21.58	**93**	**1.91**	54	1.11
FC2	123	2.5	**236**	**4.83**	**93**	**1.91**	163	3.34
FC3	317	6.49	3325	68.15	196	4.01	201	4.12
FC4	225	4.61	1828	37.46	196	4.01	152	3.12
FC5	110	2.25	3839	78.68	2688	55.09	184	3.77
FC6	249	5.10	2626	53.82	1795	36.79	127	2.6
FC7	89	1.82	3133	64.21	3551	72.78	76	1.5
FC8	**48**	**0.98**	3663	75.08	99	2.03	**23**	**0.47**
FC9	80	1.63	4111	84.25	553	11.33	51	1.04
FC10	155	3.18	3633	74.46	855	17.52	44	0.90
FC11	220	4.50	3031	62.12	242	4.96	62	1.27
Avg	162.33	3.54	2869.58	56.78	871.66	19.31	**96.66**	**2.12**

Table 6.3: Number and percentage of top journal entries marked as outlier in order to obtain a recall of 100% for **all** 7 injected outliers

	HBOS [No. Outliers]	HBOS [% Outliers]	Autoencoder [No. Outliers]	Autoencoder [% Outliers]	KNN [No. Outliers]	KNN [% Outliers]	IF [No. Outliers]	IF [% Outliers]
FC1	32	0.65	**8**	**0.16**	**5**	**0.10**	9	0.18
FC2	23	0.47	**8**	**0.16**	8	0.16	9	0.18
FC3	128	2.6	261	5.34	31	0.63	33	0.67
FC4	32	0.65	114	2.33	8	0.16	7	0.14
FC5	25	0.51	82	1.68	17	0.34	18	0.36
FC6	38	0.77	1296	26.56	28	0.57	8	0.16
FC7	89	1.82	**8**	**0.16**	7	0.14	8	0.16
FC8	**8**	**0.16**	2594	53.16	**5**	**0.10**	**5**	**0.10**
FC9	18	0.36	2545	52.16	8	0.16	6	0.12
FC10	155	3.17	2830	58.00	12	0.24	39	0.79
FC11	220	4.51	1318	27.01	11	0.22	38	0.78
Avg	64.66	1.43	1099.58	20.61	**12.72**	**0.26**	15.41	0.33

Table 6.4: Number and percentage of top journal entries marked as outlier in order to obtain a recall of 100% for the 5 **global** injected outliers

	HBOS [No. Outliers]	HBOS [% Outliers]	Autoencoder [No. Outliers]	Autoencoder [% Outliers]	KNN [No. Outliers]	KNN [% Outliers]	IF [No. Outliers]	IF [% Outliers]
FC1	284	5.82	1053	21.58	**93**	**1.91**	54	1.11
FC2	123	2.5	**236**	**4.83**	**93**	**1.91**	163	3.34
FC3	317	6.49	920	18.85	196	4.01	201	4.12
FC4	225	4.61	1828	37.46	196	4.01	152	3.12
FC5	110	2.25	3717	76.18	2688	55.09	184	3.77
FC6	249	5.10	2626	53.82	1795	36.79	127	2.6
FC7	78	1.59	3133	64.21	3551	72.78	76	1.5
FC8	**48**	**0.98**	3663	75.08	99	2.03	**23**	**0.47**
FC9	80	1.63	2509	51.43	553	11.33	51	1.04
FC10	49	1.00	3286	67.34	855	17.52	44	0.90
FC11	180	2.21	3031	62.14	242	4.96	62	1.27
Avg	143.25	3.11	2510.50	48.44	871.66	19.31	**96.66**	**2.12**

Table 6.5: Number and percentage of top journal entries marked as outlier in order to obtain a recall of 100% for the 2 **local** injected outliers

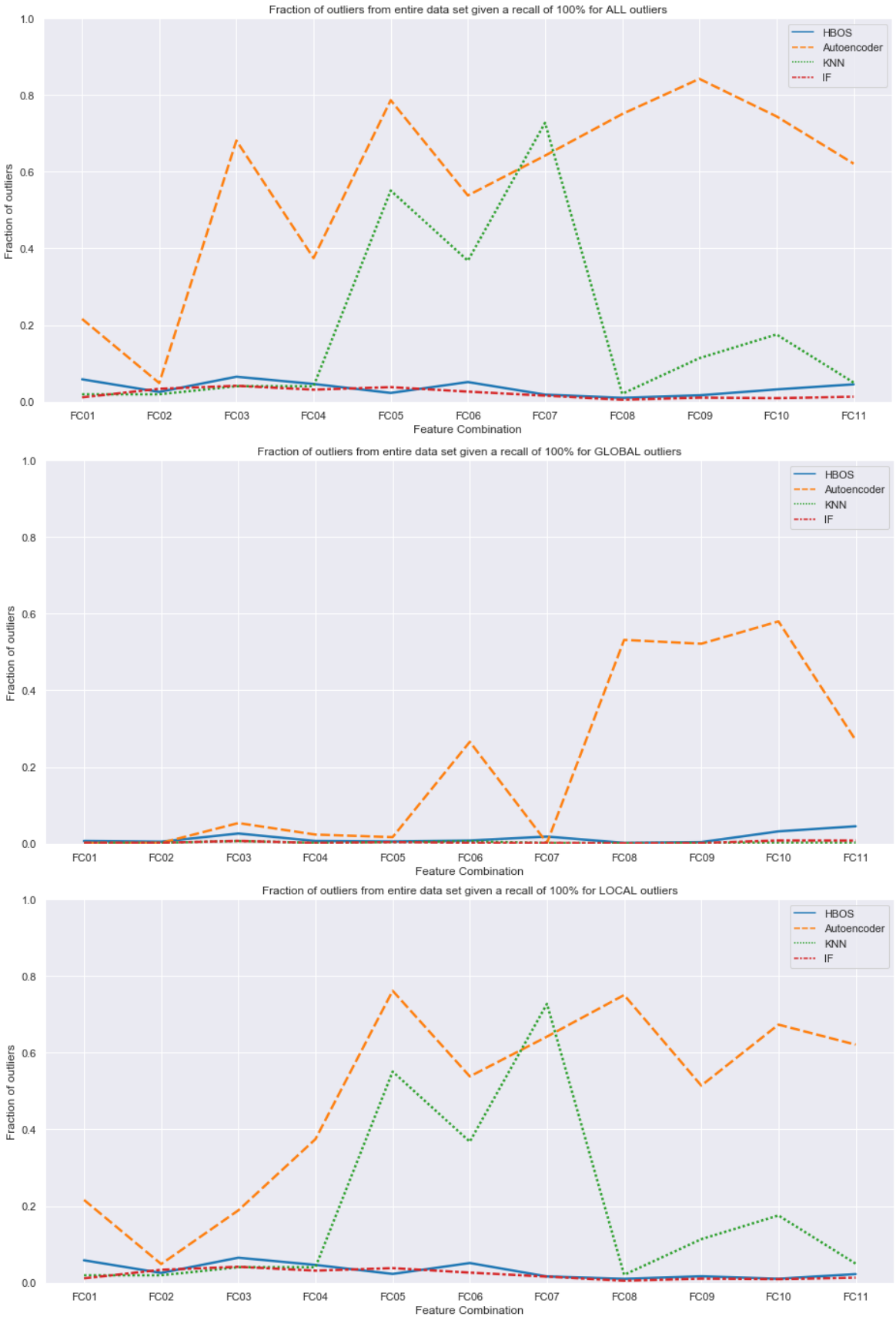

Figure 6.1: The top fraction of journal entries that have to be marked as outlier in order to obtain a recall of 100% for: *TOP*: All Synthetic Outliers, *CENTER*: Global Synthetic Outliers, *BOTTOM*: Local Synthetic Outliers

Selecting the top outliers for a recall of 100% for only global synthetic outliers it is clear that the detection performance increases, in some cases significantly, for all outlier detection techniques (table 6.4). Most interesting, on average bases, KNN is better at scoring global outliers since on average basis it is the best outlier detection algorithm only taking global outliers into account. In a comparative research by Goldstein and Uchida it was also stated that KNN is more suitable for the detection of global outliers [3].

Another interesting fact for the detection of global outliers, is that IF still performs very well and only slightly less than KNN. On average basis, IF requires the selection of the top 15 outliers and all synthetic global outliers will be included, this is only 3 more compared to KNN. For HBOS performance in terms of global outliers detected increase only on some of the feature combinations. One can see for HBOS in table 6.4 that in case of **FC10** and **FC11** performance stays the same since the same top of outlying journal entries have to be selected to include all global outliers.

Looking at the result in 6.5, only presenting the top outlying journal entries in order to include all local synthetic outliers it is clear that KNN and IF perform exactly the same as the first table. This means that for these outlier detection techniques the local outliers always have a lower outlier score than the global outliers. Therefore KNN and IF seem to be better at detecting global outliers where IF still performs very good on local outliers and even outperforming all other compared techniques. Looking at HBOS, performance slightly increases in some of the feature combinations (**FC7, FC10, FC11**). In these cases it seems that HBOS is better at detecting the local synthetic outliers than the global synthetic outliers.

Finally, the Autoencoder Neural Network seems to improve significantly for some feature combinations in case of global outliers (6.4). The Autoencoder Neural Network comes as close as equalling both KNN and IF with feature combinations **FC1, FC2, FC7**. But on average basis the Autoencder Neural Network is is underperforming to all other compared techniques, having to select averagely 1223 journal entries to include all local outliers, which is about quarter of the entire set. The same can be said about the detection rate of local synthetic outliers presented in 6.5. Although, the Autoencoder Neural Network, of all outlier detection techniques compared, improves the most when comparing the results of the local outliers to the the results of all the outliers. Where all other outlier detection techniques practically have the same result the Autoencoder Neural Network shows better results on local outliers.

6.4 Hyperparameters

The following hyperparameters have been applied in order to retrieve the best performance of each unsupervised technique taking **all** 7 synthetic outliers in considerations.

HBOS : *binwidth = static; n_bins = -1*

Autoencoder Neural Network : *Activation function hidden layers = LReLU; Learning Rate = 0.0001; Network Optimizer = Adaptive moment estimation (Adam); Network weight*

initialization = Glorot normal; Batch sizes = 128 journal entries; loss function = MSE; n_epochs = 100

KNN : *K = 10 ; method = mean*

IF : *max_features = 1.0; n_estimators=500*

These above described hyperparameters are also the same for the best performance in detecting either local or global synthetic outliers. Alongside of this the network architecture of the Autoencoder Neural Network best performance corresponding with feature combination **FC2** is as following:

Autoencoder Neural Network Architecture : [7-4-3-4-7] fully connected layers and neurons

Chapter 7

Results Case Study: Client Audit Data

Based on the results of previous the experiments presented in chapter 6, comparing four unsupervised outlier detection techniques, it was found that IF is the best performing algorithm in terms of detecting the synthetic injected outliers. Therefore this technique has been applied on two transactional data sets of two different clients from de Jong & Laan Accountants.

7.1 Client Audit Data Sets and Experiment Setup

Two transactional data sets have been used during these experiments. These data sets (table 7.1) come from two different type of clients from de Jong & Laan Accountants and are both of different size. During conduction of the experiments, these clients were being audited each by a different financial statement auditor.

The data structure of both sets are exactly the same as in the previous described experiments. Using the same procedure from previous experiments, both sets are extracted from specific client's Accounting Information System AIS in standardized XAF format and converted to tabular data.

Data Set	No. Journal Entries	No. Mutations	Business Sector	Outlying Journal Entries (threshold = 0.6)	Outlying Mutations (threshold = 0.6)
D1	15,376	342,820	Production	150 (1.0%)	323 (0.1%)
D2	8,744	40,004	Transport	51 (0.6%)	79 (0.2%)

Table 7.1: Client audit data sets and detected outliers by Isolation Forests

After applying IF on both data sets, mutations in the data set are again scored with a value from 0 to 1, indicating a scale of abnormality. A threshold at 0.6 has been set indicating all mutations having a score higher than the threshold as 'outlier'. This threshold has been set in order to be able to systematically analyse these outlying journal entries, which is done manually. These outlying mutations are analyzed by three experts of de Jong & Laan, being one of their CPAs and the two corresponding auditors of both clients. This procedure of

analyzing is visualized in figure 7.1.

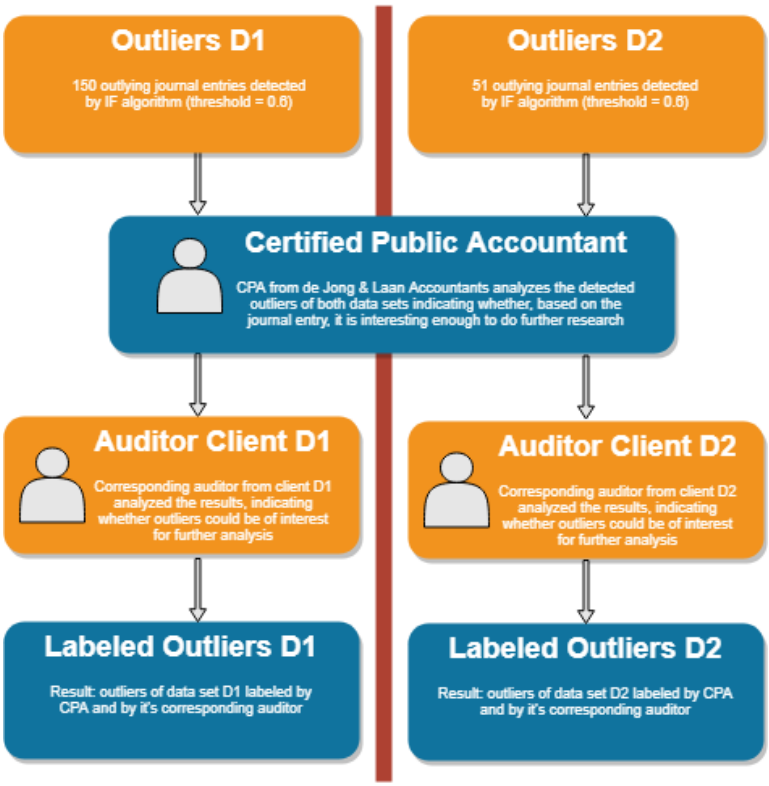

Figure 7.1: Procedure of analyzing detected outliers by IF algorithm from both client transactional data sets

Outlying journal entries, according to the IF algorithm, have been presented to a CPA of de Jong & Laan Accountant. The CPA, also being an auditing expert, indicated for each journal entry whether the journal entry is 'abnormal' enough for the CPA to perform additional research in terms of suspected errors made or an indication of fraud. The resulting outliers from the IF algorithm are therefore labeled by the CPA.

Following this procedure, the auditors of both clients also analyzed the results of the IF algorithm. This has been done after the audit process of both clients had been finished and independently of the labels indicated by the previous CPA. The auditors, indicated for each detected outlying journal entry whether they are of interest or not. Again, for a journal entry to be of interest it has to be 'abnormal' enough for the auditor to do extra research on the specific journal entry based on his specific client knowledge and audit results. The data set containing all outlying journal entries resulting from the IF algorithm is therefore again labeled by the auditor.

Besides, if the journal entry is proven to be of interest, the auditor indicated whether the specific journal entry was already checked following regular audit procedures. Resulting in an overview with all outliers labeled by a CPA and corresponding auditor.

7.2 Selected Features and Hyperparameters

During the previous experiments feature combination corresponding to the best results in terms of synthetic outliers detected is proven to be **FC8** (see table 6.2). Nonetheless during the two experiments with client audit data sets a different feature combination has been used. The feature combination that has been used on the client data is **FC11**, presented in the following table (7.2). These features have been selected because they seem less biased towards the synthetic injected outliers from previous experiments while still maintaining excellent detection performance.

Feature Name	Description	Data Type
cust_SupId	Identification of a customer or supplier to which a transaction applies to	Categorical
cas_DescDet3	Identification of standardized general ledger names	Categorical
cf_dateIndex	Number of days passed since the first transaction in the set	Numerical
cf_custMean	Mean of the amount from all transactions of a specific customer or supplier	Numerical
cf_distanceToMean	Difference between the amount of a transaction and cf_custMean	Numerical
cf_debCredAmntLog	Log transformation of the transaction amount. In case the the transaction is of type 'credit' the number is transformed to a negative number	Numerical

Table 7.2: Selected features for client cases, in previous experiments indicated as **FC11**

The following hyperparameters for IF have been used during both experiments on client audit data:

max_features = *0.75*

n_estimators = *500*

max_samples = *1.0*

bootstrap = *False*

Max_features has been set to *0.75* in order for each base estimator (tree) to sample 75% from the number of total features. The number of estimators *n_estimators* has been set to 500, therefore 500 base estimators are generated by the algorithm since this reduces the execution time of the algorithm while maintaining a good performance.

7.3 Results

Table 7.3 presents the results after evaluating the outlying journal entries identified by the IF outlier detection algorithm and an outlier threshold of 0.6. A CPA identified for both resulting sets the number of anomalous transactions as well as the corresponding auditors of both clients **D1** & **D2**.

	D1		D2	
	No.	**Perc.**	**No.**	**Perc.**
Outlying journal entries (IF)	*150*	*100.00%*	*51*	*100.00%*
Labeled as anomalous by CPA	*16*	*10.67%*	*10*	*19.91%*
Labeled as not anomalous by CPA	*134*	*89.33%*	*41*	*80.39%*
Labeled as anomalous by auditor & noticed during auditing	*53*	*35.33%*	*13*	*25.49%*
Labaled as anomalous by auditor & unnoticed during auditing	*4*	*2.97%*	*3*	*5.88%*
Labeled as anomalous by auditor total	*57*	*38.00%*	*16*	*31.37%*
Labeled as not anomalous by auditor	*93*	*62.00%*	*35*	*68.63%*

Table 7.3: Performance of IF outlier detection algorithm on both client audit data sets **D1** & **D2**, number and percentage of outliers as labeled by a CPA and auditor are given

In both cases very low fraction of outlying journal entries have been detected by the IF algorithm (**D1** $= 1.0\%$; **D2** $= 0.6\%$). In case of **D1** 16 journal entries have been labeled 'anomalous' by the CPA in comparison to 57 by the auditor. For **D2** a same pattern is clear, being that the auditor labeled more journal entries as being anomalous than the CPA, being 16 by the auditor and 10 by the CPA. In case of the auditor, precision rate for **D1** $= 38.0\%$ and for **D2** $= 31.37\%$. Taking average precision over both the CPA and the auditor results for **D1** and **D2** respectively 24.34% , 25.64%.

Most interesting is that by selecting all outlying journal entries having a higher score than 0.6 results in a small set of anomalous journal entries that haven't been noticed during the regular audit process. In case of **D1** there are 4 anomalous journal entries and 3 in case of **D2**. Both auditors indicated that these journal entries are suspicious enough from an audit perspective.

Besides investigating the detected outliers by IF, a small research has been done to the testing procedure executed by the auditors for both clients. It is analyzed which journal entries have been selected by the auditors during the audit process for further analysis. These journal entries are considered anomalous and selected based on standard audit procedures and statistical procedures.

Table 7.4 presents the journal entries that have specifically been analysed based on

regular audit procedures. What immediately stands out is that the number of selected and analyzed journal entries is less than the labeled anomalous journal entries from table 7.3. This is due to the fact that the outliers presented in table 7.4 are selected based on statistical analysis of specific subsets from the audit data set (e.g. certain ledger accounts).

Based on these results it is possible make a statement about the false negatives and the threshold that has been set during these experiments. From the 18 identified journal entries by the audit in **D1**, 8 have an outlier score higher than the threshold of 0.6 and 10 have a lower score. The average outlier score of these selected outliers is 0.594. In case of **D2** only 3 of the 21 journal entries are detected by the IF algorithm, and the average outlier score of the investigated journal entries is 0.392.

	D1	**D2**
Selected journal entries by auditor	*18*	*21*
Average IF outlier score	*0.594*	*0.392*

detected by IF (> 0.6)	*8 (44.44%)*	*3 (14.29%)*
undetected by IF (≤ 0.6)	*10 (55.56%)*	*18 (85.71%)*

Table 7.4: Selected anomalous journal entries by auditors of both clients based on regular audit procedures and their respective average outlier score and detection rate

Only taking table 7.4 into account the recall percentages are 44.44% in case of **D1** and 14.29% in case of **D2**. It is important to realise that recall percentages are only based on the journal entries found anomalous during regular audit processes. The number of these journal entries is less than the journal entries labeled as anomalous by the auditor from the IF results described earlier. Taking these into account the recall percentage will rise respectively for **D1** and **D2** to 85.07% and 88.88%.

Chapter 8

Discussion

The objective of this research has been to explore the possibilities of unsupervised outlier detection techniques during financial statement audits. In order to do so, outlier detection techniques have been identified, selected and experimented with on three different transactional audit data sets consisting of journal entries. One of the data sets contained synthetic outlying journal entries and was used to compare four selected unsupervised outlier detection techniques in terms of detection performance. One of the compared techniques was proven to be very successful and was therefore applied on two real client audit cases from de Jong & Laan Accountants.

8.1 Contributions to Research

This research makes the following contributions to the scientific field of unsupervised outlier detection.

First of all it provides a comparison of commonly, new and most promising unsupervised outlier detection techniques based on a realistic transactional audit data set. To best of knowledge, recently only a single study focused on the detection of outliers in transactional audit data consisting of journal entries with the perspective of improving the financial statement audits in terms of detection fraud en error. Besides, to the best of knowledge, this is the only study where unsupervised outlier detection techniques are applied during two live financial statement audits.

In addition, results from this study show that from the compared algorithms, IF is the best performing unsupervised outlier detection algorithm to detect injected synthetic outliers in transactional audit data. Furthermore this technique seems to be the most robust technique in terms of feature selection and hyperparameter tuning since it performs very consistent given multiple combinations of these. Aside HBOS also seems to perform well and stable in terms of detection performance, performing slightly better than IF in some situations.

Also IF seems to be applicable during financial statement audits based on qualitative feedback from auditors. IF has been applied in two live audit cases from de Jong & Laan Accountants and was in both cases able to detect interesting anomalous journal entries that haven't been identified during regular audit processes. This indicates that unsupervised out-

lier detection, and specifically IF, have potential to improve the quality of financial statement audits.

During the only recent comparable study by Schreyer [2], also applying multiple unsupervised outlier detection techniques on transactional audit data, Autoencoder Neural Networks are reported to be very successful on detection anomalous journal entries in large data sets. This research contradicts these results, whilst utilizing a very similar research method and neural network architectures. During this research Autoencoder Neural Networks have a significant worse detection performance compared to the other unsupervised techniques.

Furthermore this research expands partially on the research of Goldstein and Uchida. They concluded that KNN is a very suitable outlier detection algorithm for the detection of global outliers [3]. This can also be confirmed based on the results presented in this research, since KNN is the best algorithm in detecting the global outliers.

On a more general note, Goldstein and Uchida noted that when computation time is of importance, HBOS is the most favorable technique. Computation time is not specifically studied during this research but during experiments HBOS has been experienced as quickest followed by IF, KNN and then Autoencoder Neural Networks. Finally, this research expands on the conclusions of Appelbaum et al. [6] by expanding the field of applying analytical models during the review process of financial statement audits.

8.2 Contributions to Practice

On a practical side this thesis firstly provides a comparison of four unsupervised outlier detection techniques applied on transactional audit data with very specific synthetic outlying journal entries. Accounting firms willing to implement outlier detection techniques can easily imitate and expand this evaluation process of multiple techniques. This is relatively effortless for accounting firms located in the Netherlands since a standardized data format, utilized during this research, is available in the form of XAF audit files.

Second, this thesis provides two clear examples of how IF is effectively implemented during financial statement audits. Based on these examples accounting firms are able to implement the outlier detection technique by themselves in order to improve the audit process in terms of efficiency and effectiveness.

In case of de Jong & Laan the results of this thesis provide a clear implementation of IF. In the finalizing phase of this research IF has been implemented in the existing reports of de Jong & Laan Accountants, visualized in appendix G. From now on these existing reports, utilized during audit procedures, contain the outlier scores generated by IF for each transaction in the transactional data set. Based on a specific threshold transactions are indicated with a 'red flag' in case the outlier score is higher than the threshold (see appendix G). De Jong & Laan Accountants would be able to continuously measure performance in terms of outliers detected during financial statement audits which allows them to optimize the algorithm, for example which threshold to utilize.

8.3 Validity & Reliability

This section discusses the validity of the results of both experiments on audit data with synthetic outliers and experiments on client audit data in terms of internal and external validity.

There are some reasons to doubt the validity of the results from the comparative study of four unsupervised outlier detection techniques. In terms of internal validity the results are quite clear that based on these synthetic outliers IF has the the overall best detection performance. It has to be described that the global synthetic outlying journal entries are very anomalous journal entries and even so anomalous that they are quite unrealistic. One could argue that there is no added value for an unsupervised outlier detection algorithm to be able to detect these kind of outlying journal entries. Then again from a different perspective, it is realistic to expect that an outlier detection algorithm should at least able to detect these global outlying journal entries and that has not been the case during this research. In terms of external validity, the experiments where comparing outlier detection techniques with synthetic outliers should be easy to repeat. Although based on this research it is relatively unsure how the compared techniques perform when larger data sets are used. As described in the results, a relative small but realistic audit data set is used in which synthetic outliers have been injected. The detection performance could change when the set of journal entries is larger, it is not sure what the difference will be. Based on the results of client audit cases it is expected that there is no large difference in the size of data set, at least for the outcome of IF. This can not be said about HBOS, KNN and Autoencoder Neural Networks.

Further, during the comparative experiments it was found that Autoencoder Neural Networks have a very underwhelming detection performance. Given the good results reported by Schreyer [2] the results presented in this research are very contradictory. There has been personal contact with Schreyer about the rather disappointing results. Schreyer stated that during their research specific synthetic outliers have been injected that would be rather difficult to detect with standard statistical methods utilized during regular audit procedures. In other words, detecting outliers that are hard to find with usual 'red flag' methods has been the focus of the research from Schreyer et al. In this research, the synthetic injected outliers should be detectable utilizing standard statistical procedures. This might be a cause of the difference in terms of detection performance in this research and that of Schreyers.

On the other hand, Schreyer experimented with two very large data sets and the percentage of injected synthetic outliers is also slightly lower, being 0.06% in comparison to the 0.14% from this study. This might also be a cause of the different results, Autoencoder Neural Networks could be more suitable for larger journal entry data sets but as described before this is not studied.

For the two client audit cases there is a high confidence in terms of internal validity. IF has been applied on audit data sets from two different clients both coming from a different business sectors and having different amount of transactions and journal entries. The results have individually been evaluated by both auditors of these clients in order to identify anomalous journal entries that could be of interest from an audit perspective. A downside of this method can be that the results are analyzed shortly after the audit was executed at both

clients. The financial statements of both clients had been approved and this might have influence on the view of the evaluating auditors. Auditors could have been biased towards the fact that the financial statements were approved, indicating less anomalous journal entries resulting from IF. It could very well be that the auditors would have found more journal entries to be anomalous if the results of IF had been used before the approval of the financial statements.

Finally, this research rests on the assumption that there is a connection between anomalous / outlying journal entries and fraud or errors made during financial administration. Based on the results of this study, this can not be proven. Auditors have found journal entries in the results of the IF algorithm that are of interest from an audit perspective but there is no direct proof of errors made or fraud scenarios.

8.3.1 Summary

To summarize, the synthetic global outliers injected during the first experiments comparing multiple outlier detection techniques are so anomalous that they are quite unrealistic. One could argue what the added value would be when being able to detect these kind of outliers with an outlier detection algorithm while they can easily be found with basic statistics. Furthermore the data set used during the first experiments is relatively small and therefore detection performance on larger data sets is unknown in case of synthetic outliers. The data sets used during experiments with client data where a lot larger and there is high confidence for the presented results of the IF algorithm. A downside is that the outliers are analyzed after the audit process. It could be possible that the auditors would have found more anomalous journal entries based on the results of IF if they where analyzed during the audit process. Finally, the detection of fraud or errors made by a client still remains to be proven since the financial statements of both clients were approved.

8.4 Suggestions for Future Work

In this sections the main directions and suggestions for future work are summarized based on the results of this study.

First of all, in terms of evaluating multiple unsupervised outlier detection algorithms on transactional audit data there is a lot of room for further research. Other algorithms, also listed in section 3.3 can be applied and evaluated utilizing the same method of injecting synthetic outlying journal entries in existing audit data sets. This also serves the possibility to experiment on different kinds of audit data sets, meaning sets of different sizes and coming from different business sectors. Furthermore during this research there has been a lot of differentiation in the combinations of features used during experiments. Selection of features and creation of additional features is a field that requires more research in order to generate the most suitable data sets to perform outlier detection on.

Also evaluation of unsupervised outlier detection techniques applied on real audit cases should be research further. Only two cases have been analyzed during this study and the

results are promising in terms of detecting outliers that could be of interest for auditors. More cases need to be studied in order to assure the applicability of these techniques even more. Specifically on fraud detection which have not been proved during this study.

Lastly, during literature study interesting so called 'ensemble' outlier detection techniques are also proven to be an interesting direction. These techniques provide a more thorough list of outliers based on the average outlier score of combined outputs of multiple outlier detection techniques.

Chapter 9

Conclusion

This chapter concludes the work presented in this thesis by directly providing answers to the research questions.

9.1 Answer to Research Questions

This study aimed to investigate and explore the possibilities of unsupervised outlier detection techniques for financial statement audits. The following main research question has been followed during this research:

To what extend can unsupervised outlier detection techniques be applied to detect outliers in transactional audit data?

In order to answer the this question, experiments with unsupervised outlier detection techniques have been conducted. Experiments have been conducted on a case with realistic transactional audit data where synthetic outliers have been injected in order to measure detection performance. Followed by two live audit cases, applying unsupervised outlier detection from which results have been evaluated by three auditing experts. The main research question was divided into two sub-questions which are answered as following.

RQ1 *Which (class of) unsupervised outlier detection techniques can be applied on transactional audit data in order to detect outliers?*

Unsupervised outlier detection techniques have been identified and categorized into four different categories. These four categories are: proximity-based, subspace, statistical / probabilistic techniques and three techniques categorized as 'other'. From these categories KNN has been the chosen technique for proximity-based techniques, HBOS for statistical / probabilistic and both IF and Autoencoder Neural Networks have been selected from the remaining category. These outlier detection techniques have been applied on transactional audit data containing synthetic injected outliers. All four techniques have been able to detect outliers but on average basis KNN and Autoencoder Neural Networks perform significantly worse

than HBOS and IF. On detecting synthetic outliers, IF had the best detection rate compared to the other techniques. On average basis only the top 97 journal entries had to be marked as outlier based on outlier score in order to obtain a recall percentage of 100%. This is only 2.12% of the top scoring from the entire data set of journal entries. This is better than the other compared techniques being, HBOS, KNN and Autoencoder Neural Networks respectively requiring the $3.54\%; 19.31\%; 56.78\%$ of the top scoring journal entries from the data set to be selected for a recall of 100%. Therefore based on all synthetic injected outliers IF is the best performing unsupervised outlier detection algorithm. Taking only global outliers into consideration, KNN, slightly outperforms IF requiring the top scoring 0.26% of the data set to include all global outliers in comparison to 0.33% making KNN only suitable for the detection of global outliers. In case of local journal entries, IF is again the outlier detection algorithm of choice, requiring 2.12% of the top scoring journal entries to include all local anomalies.

This concludes that HBOS and IF both are applicable in order to detect outliers in transactional audit data.

RQ2 *Which unsupervised outlier detection technique performs best in detecting outliers that are of interest for the auditor?*

Applying IF in a live audit setting of two clients resulted in detection of outliers that are of interest for the auditor. IF labeled each journal entry from two client audit data sets with an outlier score from which all journal entries with a score higher than 0.6 have been presented to respective auditors. The auditors evaluated these journal entries by labeling them as anomalous based on possible risks from an audit perspective (fraud / error). From the first set with 51 detected outliers 16 (31.37%) were also labeled as anomalous by the auditor from which 3 (5.88%) have been fully unnoticed based on regular audit procedures. From the other client 57 (38%) have been labeled as anomalous by the auditor from which 4 (2.97%) went fully unnoticed. Both sets have also been analyzed by a CPA who indicated 16 $(10.67$%$)$ journal entries as anomalous from the 150 and 10 (19.91) from the other 50 journal entries.

Concluding that the unsupervised outlier detection technique IF performs best from all compared techniques in detecting synthetic outliers from transactional audit data sets. Also IF can be applied during financial statement audits in order to improve the quality of financial statement audits by decreasing the risk of 'missing' anomalous journal entries that could be an indication of fraud or errors made.

Bibliography

[1] A. C. Bahnsen, “Benefits of Anomaly Detection Using Isolation Forests,” 2016. [Online]. Available: https://blog.easysol.net/using-isolation-forests-anamoly-detection/

[2] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, “Detection of Anomalies in Large Scale Accounting Data using Deep Autoencoder Networks,” sep 2017. [Online]. Available: http://arxiv.org/abs/1709.05254

[3] M. Goldstein and S. Uchida, “A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data,” *PLOS ONE*, vol. 11, p. e0152173, apr 2016. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0152173

[4] ACFE, “Report to the Nation: Occupational Fraud and Abuse. Association of Certified Fraud Examiners (2018),” *Association of Certified Fraud Examiners (ACFE)*, 2018.

[5] M. van der Vegte, B. Albers, A. Bast, M. Baks, P. Dinkgreve, E. Eeftink, P. Hopstaken, M. Huisman, A. Koops, R. Lelieveld, R. Ogink, H. Renckens, and B. Wammes, “Fruade Protocol; Wat je van de controlerend accountant mag verwachten als het gaat om fraude,” Koninklijke Nederlandse Beroepsorganisatie van Accountants, Tech. Rep. December, 2018.

[6] D. A. Appelbaum, A. Kogan, and M. A. Vasarhelyi, “Analytical procedures in external auditing: A comprehensive literature survey and framework for external audit analytics,” *Journal of Accounting Literature*, vol. 40, no. January, pp. 83–101, 2018. [Online]. Available: https://doi.org/10.1016/j.acclit.2018.01.001

[7] F. E. Grubbs, “Procedures for Detecting Outlying Observations in Samples,” *Technometrics*, vol. 11, no. 1, pp. 1–21, feb 1969. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00401706.1969.10490657

[8] A. Zimek and P. Filzmoser, “There and back again: Outlier detection between statistical reasoning and data mining algorithms,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1280, nov 2018. [Online]. Available: http://doi.wiley.com/10.1002/widm.1280

[9] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, “Statistical outlier detection using direct density ratio estimation,” *Knowledge and Information*

Systems, vol. 26, no. 2, pp. 309–336, feb 2011. [Online]. Available: http://link.springer.com/10.1007/s10115-010-0283-2

[10] C. C. Aggarwal, *Outlier Analysis*. Cham: Springer International Publishing, 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-47578-3

[11] F. A. Amani and A. M. Fadlalla, "Data mining applications in accounting: A review of the literature and organizing framework," *International Journal of Accounting Information Systems*, vol. 24, pp. 32–58, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.accinf.2016.12.004

[12] Price Waterhouse Coopers, "Understanding a financial statement audit," Price Waterhouse Coopers, Tech. Rep. January 2013, 2013. [Online]. Available: https://www.pwc.com/im/en/services/Assurance/pwc-understanding-financial-statement-audit.pdf

[13] International Accounting Standards Board, "IASB clarifies its definition of 'material'." [Online]. Available: https://www.ifrs.org/news-and-events/2018/10/iasb-clarifies-its-definition-of-material/

[14] B. Dikmen and G. Küçükkocaolu, "The detection of earnings manipulation: the three-phase cutting plane algorithm using mathematical programming," *Journal of Forecasting*, vol. 29, no. 5, pp. 442–466, 2010. [Online]. Available: http://doi.wiley.com/10.1002/for.1138

[15] R. S. Debreceny and G. L. Gray, "Data mining journal entries for fraud detection: An exploratory study," *International Journal of Accounting Information Systems*, vol. 11, no. 3, pp. 157–181, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.accinf.2010.08.001

[16] F. Benford, "The Law of Anomalous Numbers," *Proceedings of the American Philosophical Society*, vol. 78, no. 4, pp. 551–572, 1938.

[17] D. McGilvray, *Executing data quality projects: Ten steps to quality data and trusted information (TM)*. Elsevier, 2008.

[18] M. Bouguessa, "A practical outlier detection approach for mixed-attribute data," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8637–8649, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2015.07.018

[19] SRA; Nederlandse Belastingdienst, "XML Platform," 2014. [Online]. Available: https://www.softwarepakketten.nl/pag_reg/54&mnreg=175&bronw=6/auditfile_organisatie_contact.htm

[20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, dec 2008, pp. 413–422. [Online]. Available: https://r-forge.r-project.org/projects/iforest/http://ieeexplore.ieee.org/document/4781136/

[21] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python Toolbox for Scalable Outlier Detection," *arXiv preprint arXiv:1901.01588*, pp. 1–6, jan 2019. [Online]. Available: http://arxiv.org/abs/1901.01588

[22] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," in *KI-2012: 35th German Conference on Artificial Intelligence*, vol. 1, 2012, pp. 59–63. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.401.5686&rep=rep1&type=pdf

[23] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, "YALE," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, vol. 1, no. 2. New York, New York, USA: ACM Press, 2006, p. 935. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1150402.1150531

[24] B. Kitchenham, "Procedures for Performing Systematic Literature Reviews," Keele University, Keele, Tech. Rep., 2004. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Procedures+for+Performing+Systematic+Literature+Review+in+Software+Engineering#1

[25] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and Software Technology*, vol. 106, no. May 2018, pp. 101–121, 2019. [Online]. Available: https://doi.org/10.1016/j.infsof.2018.09.006

[26] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, jul 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1541880.1541882

[27] D. Wit, "Data-driven audit with anomaly detection algorithms an explorative study about the application of unsupervised machine learning to detect exceptions in transaction level audit data," mastersthesis, Eindhoven University of Technology, 2016.

[28] M. Albashrawi, "Detecting financial fraud using data mining techniques: a decade review from 2004 to 2015," *Journal of Data Science*, vol. 14, no. 3, pp. 553—-569, dec 2016. [Online]. Available: http://www.jds-online.com/file_download/558/10-Detecting+Financial+Fraud+Using+Data+Mining+Techniques-JDS_V3.pdf

[29] C. C. Aggarwal, *Data Mining - Chapter 8: Outlier Detection*. Springer International Publishing, 2015. [Online]. Available: http://link.springer.com/10.1007/978-3-319-14142-8

[30] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*, 2000, pp. 427–438. [Online]. Available: http://portal.acm.org/citation.cfm?doid=342009.335437

[31] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF,” *SIGMOD ’00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, vol. 29, no. 2, pp. 93–104, jun 2000. [Online]. Available: http://portal.acm.org/citation.cfm?doid=335191.335388

[32] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, “LoOP,” in *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM ’09*. New York, New York, USA: ACM Press, 2009, pp. 1649–1652. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1645953.1646195

[33] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos, “LOCI: fast outlier detection using the local correlation integral,” in *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*. IEEE, 2002, pp. 315–326. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a461085.pdfhttp://ieeexplore.ieee.org/document/1260802/

[34] Z. Huang, “A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining,” *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pp. 1–8, 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=D76EC3805AAB98C839B58E7C256676C9?doi=10.1.1.6.4718&rep=rep1&type=pdf

[35] C. Angiulli, Fabrizio and Pizzuti, “Fast Outlier Detection in High Dimensional Spaces,” in *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*. Springer-Verlag, 2002, pp. 15—-26. [Online]. Available: http://dl.acm.org/citation.cfm?id=645806.670167

[36] N. N. Suri, M. N. Murty, and G. Athithan, “Detecting outliers in categorical data through rough clustering,” *Natural Computing*, vol. 15, no. 3, pp. 385–394, 2016.

[37] M. L. Shyu, S. C. Chen, K. Sarinnapakorn, and L. Chang, “A novel anomaly detection scheme based on principal component classifier,” University of Miami, Miami, Tech. Rep., 2003.

[38] J. Ma and S. Perkins, “Time-series novelty detection using one-class support vector machines,” in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 3. IEEE, 2003, pp. 1741–1745. [Online]. Available: http://ieeexplore.ieee.org/document/1223670/

[39] V. Hodge and J. Austin, “A Survey of Outlier Detection Methodologies,” *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, oct 2004. [Online]. Available: http://link.springer.com/10.1023/B:AIRE.0000045502.10941.a9

[40] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, “A review of novelty detection,” *Signal Processing*, vol. 99, pp. 215–249, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2013.12.026

[41] S. Sajidha, S. P. Chodnekar, and K. Desikan, "Initial seed selection for K-modes clustering A distance and density based approach," *Journal of King Saud University - Computer and Information Sciences*, may 2018. [Online]. Available: https://doi.org/10.1016/j.jksuci.2018.04.013https://linkinghub.elsevier.com/retrieve/pii/S1319157818300065

[42] M. Goyal, "A Review on K-Mode Clustering Algorithm," *International Journal of Advanced Research in Computer Science*, vol. 5, no. 7, pp. 725–729, aug 2017. [Online]. Available: http://ijarcs.info/index.php/Ijarcs/article/view/4301

[43] C. C. Aggarwal, *Data Mining - Chapter 9: Advanced Outlier Detection*. Springer International Publishing, 2015. [Online]. Available: http://link.springer.com/10.1007/978-3-319-14142-8

[44] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996. [Online]. Available: https://www.mimp.gob.pe/adultomayor/regiones/Lima_Metro2.htmlhttp://ieeexplore.ieee.org/document/543975/

[45] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.09.037

[46] B. Blumberg, D. R. Cooper, and P. S. Schindler, *Business Research Methods*, 4th ed. Mcgraw-Hill Education - Europe, 2014.

[47] R. J. Wieringa, *Design science methodology: For information systems and software engineering*. Springer Heidelberg New York Dordrecht London, 2014.

[48] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," pp. 25–36, 2013.

[49] A. van Wyk, "Encoding Cyclical Features for Deep Learning," 2018. [Online]. Available: https://www.avanwyk.com/encoding-cyclical-features-for-deep-learning/

Appendix A

XAF Audit File Model & Database Diagram XAF Auditfile

The following model represents the structure of XAF audit files, the structure of the data sets utilized during experiments conducted in this research.

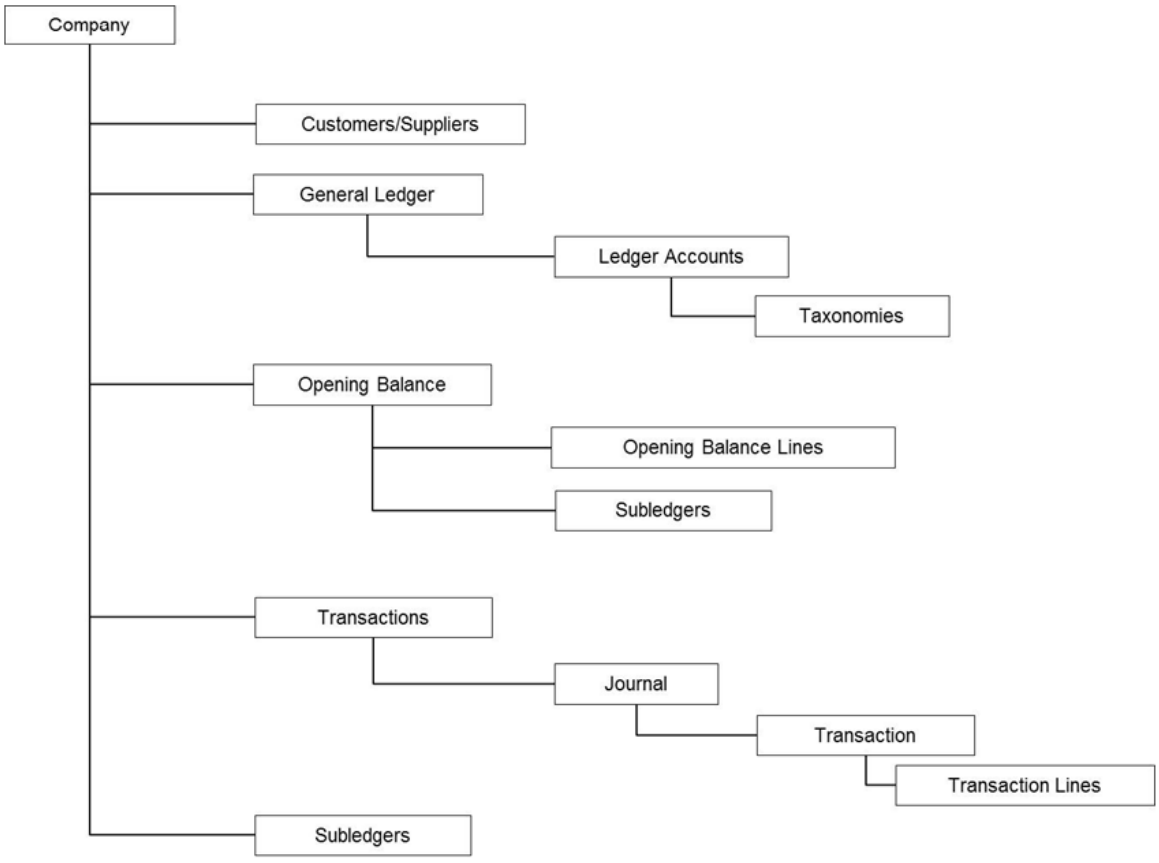

Figure A.1: XAF transaction model [19]

The following database diagram visualizes relationships between the relevant tables. These tables are the result of preprocessing XAF files into a tabled structure, which is in fact a relational database. Based on the relations a single data set can be created containing of journal entries and relevant attributes.

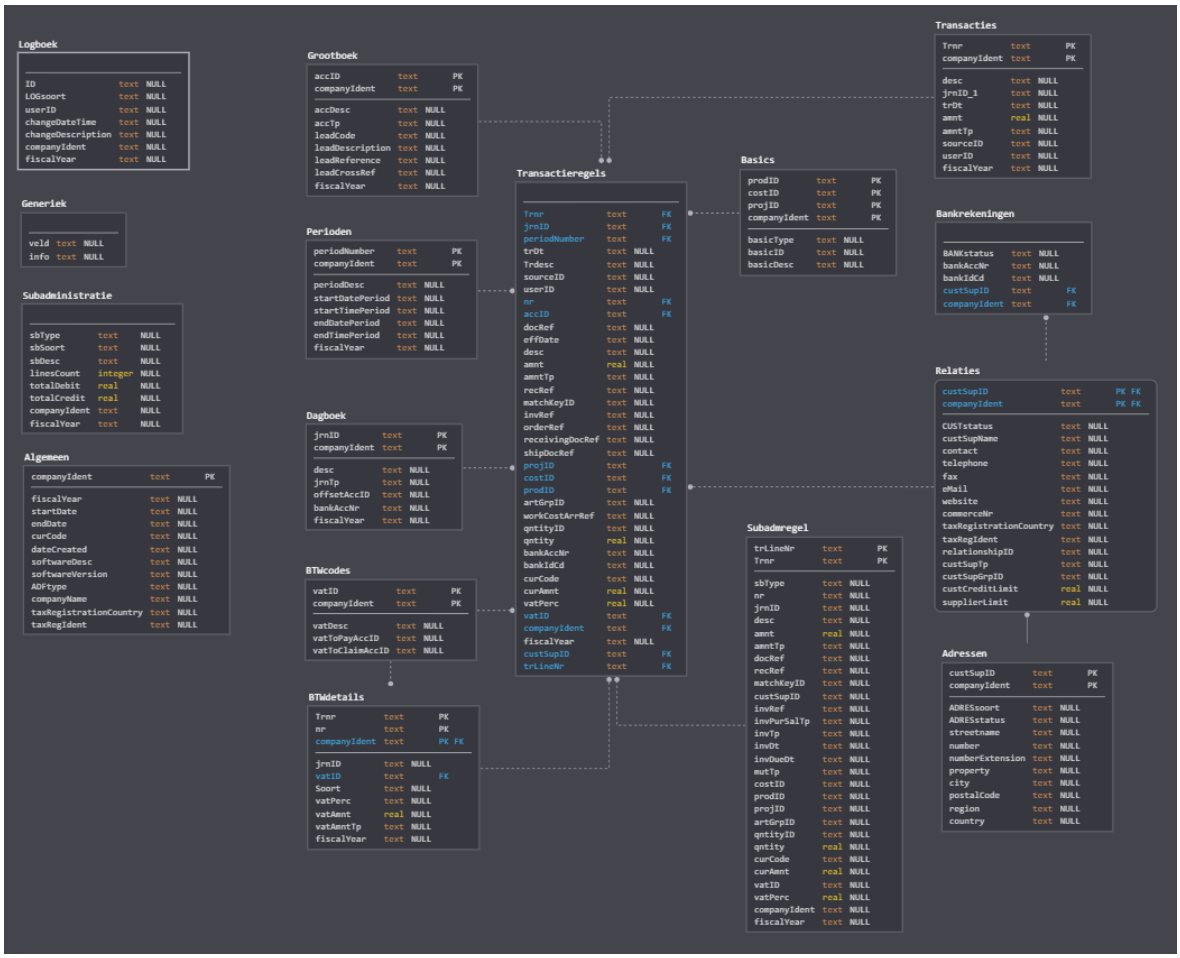

Figure A.2: Database diagram XAF auditfile

Appendix B

Transactional Audit File Sample

Table B.1 represents a single journal entry from a transactional audit file. Journal entries are indicated with an ID *(tran_Nr)* from which each mutation is indicated with a line number *(rgl_Nr)*. The date on which the journal entry has been created is given *(rgl_trDt)* including with the period when it took place being a number indicating the month or quarter *(rgl_periodNumber)*. Furthermore the year in which the transaction took place is indicated *(rgl_fiscalYear)* along with an identification of a customer or supplier and their country *(cust_SupId, cust_SupName, cust_taxRegistrationCountry)*. Finally the identification number of both the journal *(dgb_jrnId)* and the ledger account *(gb_accId)* is given for each mutation.

tran_Nr	rgl_nr	rgl_trDt	rgl_period Number	rgl_amt	rgl_amntTp	rgl_fiscalYear	cust_SupId	cust_SupName	cust_tax RegistrationCountry	dgb_jrnId	gb_accId
17315	0	3-1-2017	1	625.34	C	2017				2200	2200
17315	1	3-1-2017	1	445.92	D	2017	c1233	Customer Z	NL	2200	1500
17315	2	3-1-2017	1	98.36	D	2017	c1233	Customer Z	NL	2200	1500
17315	3	3-1-2017	1	81.06	D	2017	c1233	Customer Z	NL	2200	1500

Table B.1: Example journal entry from transactional audit data set

Appendix C

Engineered Features

Sine and cosine transformations of day, weekday, month and dateindex; The idea of transforming the transaction date, being cyclical data, comes from a post on Kaggle posted by Van Wyk [49]. The date of a transaction is a cyclical feature meaning that the feature occurs in specific cycles. Most common cyclical features are months, days, weekdays, hours, minutes and seconds. The reason these features are encoded is to make it clear to deep learning algorithms that the features occur in cycles.

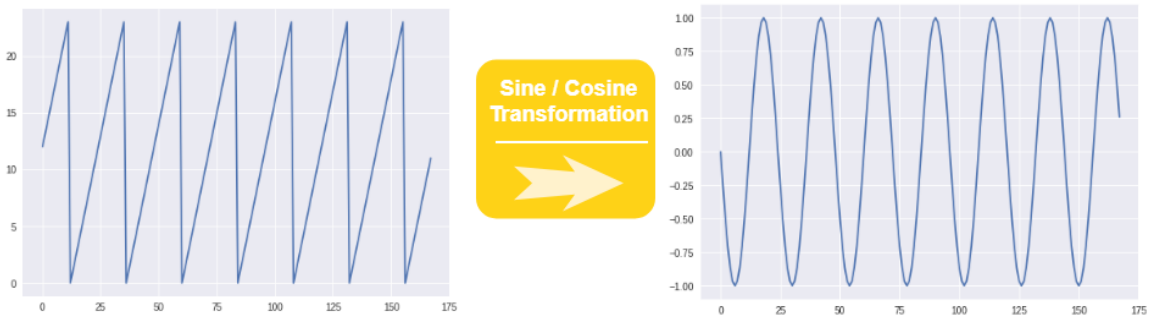

Figure C.1: Sine transformation of 7 days hourly data

Figure C.1 visualizes the transformation of each hour on a given day in a week, a cycle between 0 and 24, repeating 7 times. The graph on left side illustrates the problem with presenting cyclical data, being the discontinuities in the graph at the end of each cycle when the hour value overflows to 0. In the figure it is clearly visible that on the right side the absolute difference between 22:00 and 23:00 is 1. When considering 23:00 and 00:00 a jump occurs even though the difference is only one hour [49]. Applying the following two formulas transform a single feature, containing values between 0 and 24, into two features from which the sine transformation is plotted on the right side of figure C.1. This problem also occurs at the end of each month (31 to 1), day of year (365 to 1), minute (60 to 0) and so on.

$$x_{sin} = \sin\left(\frac{2 * \pi * x}{\max(x)}\right) \tag{C.1}$$

$$x_{cos} = \cos\left(\frac{2 * \pi * x}{\max(x)}\right) \tag{C.2}$$

Plotting both dimensions X_{sin} and X_{cos} results in a perfect cycle visualized in figure C.2. In this figure all 24 hours of a day are visible and the distance between each hour is the same.

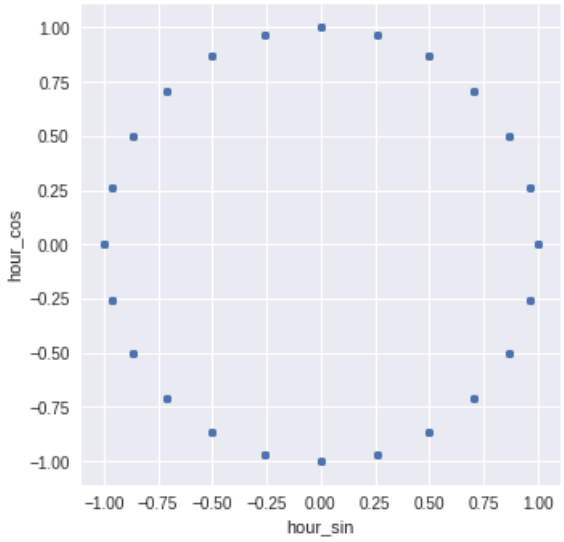

Figure C.2: Plot x = Sine / y = Cosine

cf_debCredAmnt, cf_debCredAmntLog; cf_debCredAmnt has been generated in order to make a numerical distinction between a debit or credit transaction. In case a mutation is of type 'credit' the transaction amount is transformed to a negative number. A debit mutation is an accounting entry that either increases an asset or expense account, or decreases a liability or equity account. It is positioned to the left in a balance sheet. A credit is an accounting entry that either increases a liability or equity account, or decreases an asset or expense account. It is positioned to the right in a balance sheet.

cf_debCredAmntLog, see third plot in figure C.3, describe the same feature but before transforming the mounts to either negative or positive the amounts are transformed by a natural logarithm. The reason of this is that the amount of the mutations is heavily skewed to the right. Applying the natural logarithm to the data set solves this problem, visualized in the second plot of figure C.3.

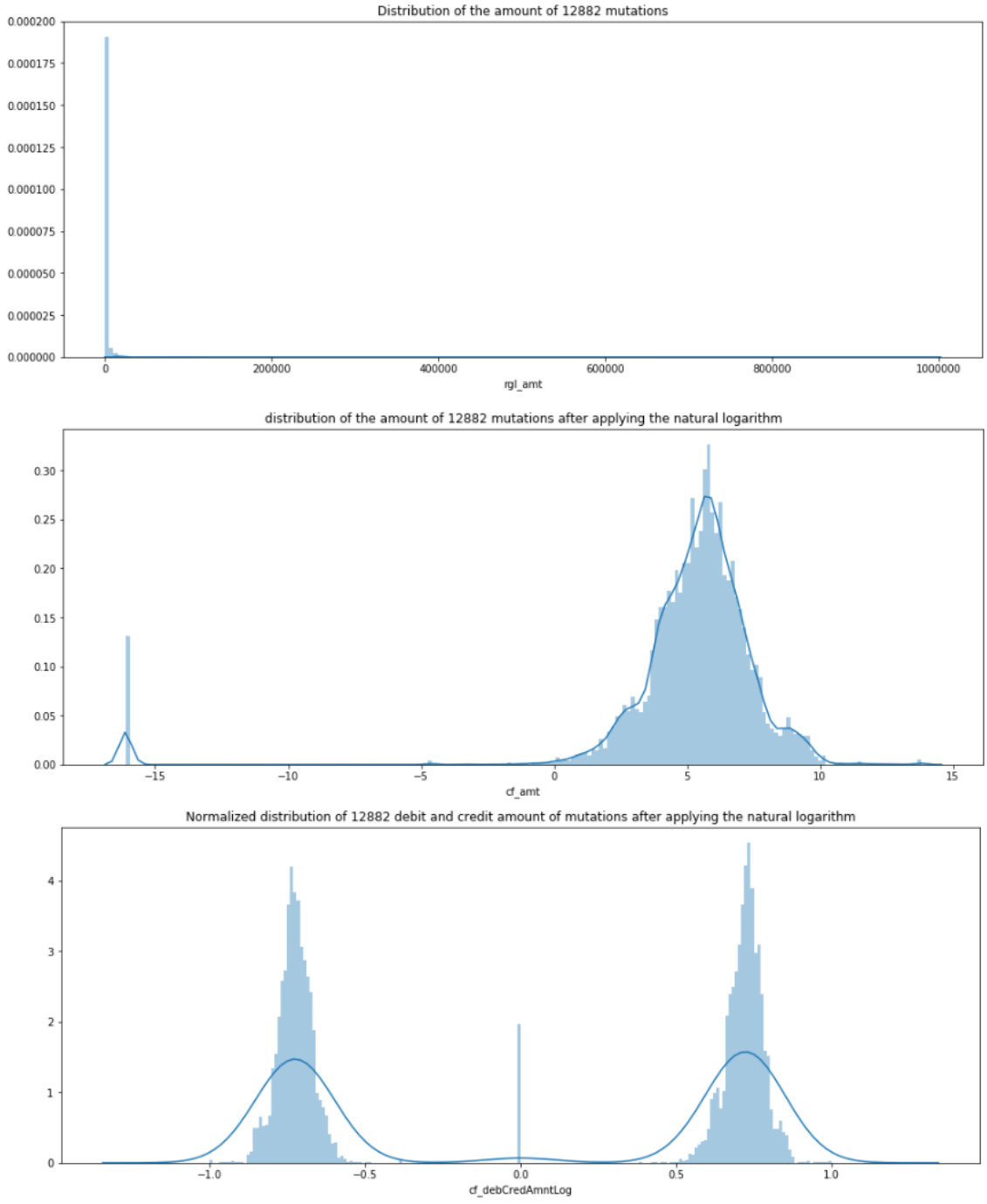

Figure C.3: Distribution of amount before and after natural logarithm scaling

Appendix D

Synthetic Outlier Experiment Results, Feature Combinations and Hyperparameter Combinations

In this appendix the full results of the conducted experiments with HBOS, IF, KNN and Autoencoder Neural Networks are presented in this chapter. The results for each feature combination along with the used parameters are presented in tables following: HBOS table D.1, IF table D.2, KNN table D.3, Autoencoder Neural Networks D.4.

Feature Combinations	binwidth	n_bins	Fraction of top n outliers	Top n outliers for recall 100%
F1	static	-1	0,351966262	284
F2	static	-1	0,321527495	123
F3	static	-1	0,241763168	317
F4	static	-1	0,39215112	225
F5	static	-1	0,38121547	110
F6	static	-1	0,381016043	249
F7	static	-1	0,411327762	89
F8	static	-1	0,315087829	48
F9	static	-1	0,338110902	80
F10	static	-1	0,205278822	155
F11	static	-1	0,362022854	220
F12	static	-1	0,315089949	48

Table D.1: Histogram-Based Outlier Score, feature combinations tested and hyperparameters tested with corresponding results in terms of top outliers required to be selected for a 100% recall

Feature Combination	max_features	n_estimators	Top n outliers for recall 100%	Fraction of top n outliers
F1	0.75	100	153	0.43789478
F1	0.75	250	141	0.427677493
F1	0.75	500	112	0.454093188
F1	1.0	100	54	0.58178571
F1	1.0	250	63	0.571255002
F1	1.0	500	69	0.562932092
F2	0.75	100	186	0.366121537
F2	0.75	250	177	0.353375962
F2	0.75	500	182	0.347666693
F2	1.0	100	163	0.379990349
F2	1.0	250	176	0.370717872
F2	1.0	500	166	0.388195782
F3	0.75	100	305	0.325180324
F3	0.75	250	289	0.324418247
F3	0.75	500	272	0.333412694
F3	1.0	100	201	0.404213067
F3	1.0	250	258	0.374135923
F3	1.0	500	251	0.380905246
F4	0.75	100	412	0.339305758
F4	0.75	250	346	0.365371176
F4	0.75	500	282	0.390802535
F4	1.0	100	152	0.526946574
F4	1.0	250	162	0.520686795
F4	1.0	500	171	0.509843611
F5	0.75	100	321	0.288316992
F5	0.75	250	284	0.301806207
F5	0.75	500	268	0.308562478
F5	1.0	100	184	0.355459487
F5	1.0	250	271	0.310433077
F5	1.0	500	226	0.32621944
F6	0.75	100	127	0.441553669
F6	0.75	250	210	0.398092016
F6	0.75	500	196	0.430623707
F6	1.0	100	170	0.431084222
F6	1.0	250	208	0.403290016
F6	1.0	500	176	0.419805043
F7	0.75	100	76	0.495519838
F7	0.75	250	80	0.493373087
F7	0.75	500	105	0.475198425
F7	1.0	100	142	0.430459133
F7	1.0	250	103	0.467118215
F7	1.0	500	118	0.453401693
F8	0.75	100	26	0.414387003
F8	0.75	250	25	0.416009269
F8	0.75	500	26	0.419201719
F8	1.0	100	24	0.523508665
F8	1.0	250	24	0.509562091
F8	1.0	500	23	0.511838359
F9	0.75	100	51	0.49195008
F9	0.75	250	53	0.492707907
F9	0.75	500	51	0.493230354
F9	1.0	100	51	0.504126507
F9	1.0	250	54	0.507700789
F9	1.0	500	52	0.505362175
F10	0.75	100	44	0.466545069
F10	0.75	250	57	0.455743933
F10	0.75	500	58	0.452349741
F10	1.0	100	55	0.496138627
F10	1.0	250	55	0.499272193
F10	1.0	500	59	0.482807253
F11	0.75	100	75	0.451984292
F11	0.75	250	66	0.47820066
F11	0.75	500	63	0.495704281
F11	1.0	100	68	0.492789922
F11	1.0	250	63	0.510001757
F11	1.0	500	62	0.513407844

Table D.2: Isolation Forests, feature combinations tested and hyperparameters tested with corresponding results in terms of top outliers required to be selected for a 100% recall, encoding = labelencoder

Feature Combination	k	Fraction of top n outliers	Top n outliers for recall 100%
F1	10	0,010658709	93
F1	25	0,010892455	218
F1	50	0,010884947	742
F1	100	0,070883003	2874
F1	150	0,159882607	2529
F2	10	0,013599011	93
F2	25	0,012615488	207
F2	50	0,012337885	595
F2	100	0,031783035	2552
F2	150	0,059900539	2338
F3	10	0,004221138	196
F3	25	0,003901177	1074
F3	50	0,064667403	2853
F3	100	0,157087539	2717
F3	150	0,131586883	2957
F4	10	0,005267358	196
F4	25	0,005372844	1074
F4	50	0,089079492	2773
F4	100	0,185525342	3800
F4	150	0,155616833	4338
F5	10	0,052293483	4260
F5	25	0,058419486	4277
F5	50	0,096957579	4225
F5	100	0,125731416	4326
F5	150	0,105885428	4416
F6	10	0,127745995	4249
F6	25	0,120394895	4197
F6	50	0,122683775	4191
F6	100	0,101392428	4390
F6	150	0,083903399	4456
F7	10	0,366449117	3760
F7	25	0,33879359	3717
F7	50	0,261628147	3798
F7	100	0,187687295	3900
F7	150	0,132374673	4052
F8	10	0,007540648	3092
F8	25	0,006120247	3801
F8	50	0,005192838	4015
F8	100	0,006740529	4074
F8	150	0,045054894	4021
F9	10	0,003639375	3667
F9	25	0,00367588	4162
F9	50	0,054717318	3897
F9	100	0,081487981	4177
F9	150	0,068033899	4403
F10	10	0,00317783	3786
F10	25	0,003424563	4216
F10	50	0,054578813	3899
F10	100	0,053623237	4276
F10	150	0,045396907	4415
F11	10	0,011399841	3049
F11	25	0,009308828	4017
F11	50	0,058090358	4044
F11	100	0,056066226	4355
F11	150	0,048069029	4454

Table D.3: K-Nearest Neighbor, feature combinations tested and hyperparameters tested with corresponding results in terms of top outliers required to be selected for a 100% recall, encoding = labelencoder

Feature Combination	Epochs	Loss	Batch-size	Activation Function	Weight Init	Architecture	Top n outliers for recall 100%	Fraction of top n outliers
F1	50	Cross-entropy	128	LReLu	Glorot	9-8-4-3-4-8-9	1053	0,14276099
F2	50	Cross-entropy	128	LReLu	Glorot	7-4-3-4-7	4477	0,14172798
F3	50	Cross-entropy	128	LReLu	Glorot	17-16-8-4-3-4-8-16-17	3325	0,19375917
F4	50	Cross-entropy	128	LReLu	Glorot	11-8-4-3-4-8-11	1828	0,13266159
F5	50	Cross-entropy	128	LReLu	Glorot	762-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-762	3839	0,09931865
F6	50	Cross-entropy	128	LReLu	Glorot	764-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-764	4164	0,15384644
F7	50	Cross-entropy	128	LReLu	Glorot	733-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-733	4067	0,17705450
F8	50	Cross-entropy	128	LReLu	Glorot	724-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-764	3663	0,03999348
F9	50	Cross-entropy	128	LReLu	Glorot	752-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-752	4259	0,04698509
F10	50	Cross-entropy	128	LReLu	Glorot	751-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-751	4488	0,03062418
F11	50	Cross-entropy	128	LReLu	Glorot	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750	4141	0,08445168
F7	100	Cross-entropy	128	LReLu	Glorot	733-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-733	3133	0,10766981
F9	100	Cross-entropy	128	LReLu	Glorot	752-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-752	4879	0,01404615
F10	100	Cross-entropy	128	LReLu	Glorot	751-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-751	4583	0,01698320
F11	100	Cross-entropy	128	LReLu	Glorot	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750	3256	0,03817647
F9	500	Cross-entropy	128	LReLu	Glorot	752-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-752	4874	0,04913689
F10	500	Cross-entropy	128	LReLu	Glorot	751-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-751	3633	0,06613985
F11	500	Cross-entropy	128	LReLu	Glorot	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750	4830	0,01392878
F9	50	Cross-entropy	128	LReLu	Glorot	752-512-256-128-64-40-64-128-256-512-752	4879	0,10967398
F10	50	Cross-entropy	128	LReLu	Glorot	751-512-256-128-64-40-64-128-256-512-751	4869	0,14446293
F11	50	Cross-entropy	128	LReLu	Glorot	750-512-256-128-64-40-64-128-256-512-750	4868	0,15333011
F11	100	Combined Cross-entropy and Mean Squared Error	128	LReLu	Glorot	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750	3031	0,03864413
F11	500	Combined Cross-entropy and Mean Squared Error	128	LReLu	Glorot	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750	3192	0,02003890
F1	20	Cross-entropy	128	LReLu	Glorot	9-8-4-3-4-8-9	2310	0,08508960
F2	20	Cross-entropy	128	LReLu	Glorot	7-4-3-4-7	4527	0,19124886
F3	20	Cross-entropy	128	LReLu	Glorot	17-16-8-4-3-4-8-16-17	3379	0,18085662
F4	20	Cross-entropy	128	LReLu	Glorot	11-8-4-3-4-8-11	3356	0,14229643
F5	20	Cross-entropy	128	LReLu	Glorot	762-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-762	4678	0,19878688
F6	20	Cross-entropy	128	LReLu	Glorot	764-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-764	4342	0,18247280
F7	20	Cross-entropy	128	LReLu	Glorot	733-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-733	3979	0,42184192
F8	20	Cross-entropy	128	LReLu	Glorot	724-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-764	4217	0,13885471
F9	20	Cross-entropy	128	LReLu	Glorot	752-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-752	4576	0,01883057
F10	20	Cross-entropy	128	LReLu	Glorot	751-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-751	4722	0,07042647
F11	20	Cross-entropy	128	LReLu	Glorot	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750	4576	0,07276532
F1	100	Mean Squared Error	128	LReLu	Glorot	9-8-4-3-4-8-9	2174	0,08406742
F2	100	Mean Squared Error	128	LReLu	Glorot	7-4-3-4-7	236	0,00008460
F3	100	Mean Squared Error	128	LReLu	Glorot	17-16-8-4-3-4-8-16-17	3711	0,03047617
F4	100	Mean Squared Error	128	LReLu	Glorot	11-8-4-3-4-8-11	2948	0,07215566
F5	100	Mean Squared Error	128	LReLu	Glorot	762-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-762	4306	0,07732124
F6	100	Mean Squared Error	128	LReLu	Glorot	764-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-764	2626	0,21509249
F7	100	Mean Squared Error	128	LReLu	Glorot	733-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-733	3917	0,45835110
F8	100	Mean Squared Error	128	LReLu	Glorot	724-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-764	4085	0,00210724
F9	100	Mean Squared Error	128	LReLu	Glorot	752-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-752	4111	0,00800565
F10	100	Mean Squared Error	128	LReLu	Glorot	751-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-751	4159	0,01245258
F11	100	Mean Squared Error	128	LReLu	Glorot	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750	4211	0,01455047
F11	1000	Cross-entropy	128	LReLu	Glorot	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750	4612	0.007794081
F11	1000	Cross-entropy	128	LReLu	Glorot	750-512-256-128-64-40-30-40-64-128-256-512-750	4876	0.13303205423643105

Table D.4: Autoencoder, feature combinations tested and hyperparameters tested with corresponding results in terms of top outliers required to be selected for a 100% recall

Appendix E

Architectures Autoencoder Neural Network

Autoencoder Neural Network experiments have been generated based on the work of Schreyer et al. [2]. Initially the same architecture has been implemented as Schreyer et al., using the Python deep learning library *Keras* running on top of *Tensorflow 1.13.1*. The following fixed hyperparameters have been used during experimenting (for more detail see [2]):

Activation function hidden layers : Leaky rectified linear units (LReLU) with a scaling factor of $\alpha = 0.4$

Learning rate : Equal learning rate applied of $\eta = 0.0001$

Network optimizer : Adam

Network weight initialization : Glorot normal

Batch sizes : Mini batch sizes of 128 data points has been used

The categorical features of the data set have been encoded to a numerical value by using one-hot encoding. Therefore, for each unique value a dimension is added to the data set, resulting into very high dimensions in case of some feature combinations. Varying dimensions resulted into varying architectures that have been experimented with. Table E.1 presents some of the architectures experimented with for some of the feature combinations given one-hot encoding for categorical variables. Along with one-hot encoding the numerical features are normalized using min-max normalization resulting in only values between 0 and 1 for the entire data set.

	No. Features	No. Dimensions Encoded	Fully Connected Layers and Neurons
FC1	*9*	*9*	9-8-4-3-4-8-9
FC4	*10*	*11*	11-8-4-3-4-8-11
FC11	*6*	*750*	750-512-256-128-64-32-16-8-4-3-4-8-16-32-64-128-256-512-750

Table E.1: Examples of network architecture experimented with corresponding to feature combination

Due to the fact that feature combination **FC11** contains a lot of categorical values the encoded dimension of the set becomes very high. The presented architecture are basically

the same as the best architecture presented in the work of Schreyer et al., the only difference being the number of input and output dimensions. In case of **FC11** the input dimension is 750 where Schreyer et al. only experimented with input dimensions of 401 and 576 [2].

The Autoencoder Neural Network is trained in order to reconstruct it's input. Network training is typically done by minimizing a loss function. Initially the same loss function, being the cross-entropy loss, as Schreyer et al. is used during experimenting and is given by:

$$\mathcal{L}_\theta(x^i;\hat{x}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} x_j^n ln(\hat{x}_j^n) + (1-x_j^n)ln(1-\hat{x}_j^n) \tag{E.1}$$

For a set of *n* data point (mutations) x^i and their respective reconstructions $\hat{x}^i$ over all mutation features j = 1, ..., k. During these experiments the training batch-size was kept constant so therefore n = 128 and in case of **FC11** the number of features would be 750. This loss function is typical in neural networks in a classification model whose output is a probability between 0 and 1. For the encoded categorical variables, according to Schreyer et al., the loss function $\mathcal{L}_\theta(x^i;\hat{x})$ measures the deviation between two independent Bernoulli distributions [2]. Crucial flaw is that this loss function is well optimized for Bernoulli distributions being the one-hot encoded categorical variables in the data set but not for the continuous numerical variables in the data set. The continuous variables in the data set can be any value between 0 and 1 whereas the categorical variables in the data set can only be 0 or 1. Utilizing $\mathcal{L}_\theta(x^i;\hat{x})$ results into the fact that the continuous vales are treated as a probability while this is not the case.

This flaw has been addressed in a conversation with M.Schreyer, one of the authors of the paper cited before. During this conversation it was stated that, ideally the loss function should be combined with a loss function suitable for regression problems. An example of a loss function well known in regression problems is Mean Squared Error MSE. In case of *n* mutations and corresponding mutation features *j*, MSE can be given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}(x_j^n - \hat{x}_j^n)^2 \tag{E.2}$$

Following rather disappointing results, presented in the following chapter, experiments with cross-entropy loss and MSE have been conducted. Also a combined loss function has been implemented during experiments where the loss for categorical variables is calculated by cross-entropy and for continuous variables by MSE. This loss function is given by:

$$\mathcal{LM}_\theta(x^i;\hat{x}) = \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=1}^{k} x_j^n ln(\hat{x}_j^n) + (1-x_j^n)ln(1-\hat{x}_j^n) + \sum_{l=1}^{m}(x_l^n - \hat{x}_l^n)^2\right] \tag{E.3}$$

Here cross-entropy is calculated over all one-hot encoded categorical features j = 1, ..., k of a given mutation *n*. MSE is calculated over continuous numerical features l = 1, ..., m for a given mutation *n*. Both results are summed and the average is taken over n mutations in a batch.

Aside from different loss function the Autoencoder Neural Networks has been trained on, different architectures have been experimented with. These architectures are presented in appendix E. Finally different number of epochs has been used in order to optimize the performance of the Autoencoder Neural Network

Appendix F

Synthetic Injected Outlying Journal Entries

The tables presented in this section represent 7 injected synthetic anomalous journal entries in the transactional data set used during experiments comparing four unsupervised outlier detection techniques. Comparison is made based on the detection rate of these journal entries. Finally, in compliance with data privacy regulations, the journal entry attributes have been anonymized.

tran_Nr	**rgl_trDt**	**rgl_amt**	**rgl_amntTp**	**cust_SupId**	**cust_SupName**	**dgb_jrnId**	**gb_accId**
19420	2017-04-07	12,345.67	C			1500	1500
19420	2017-04-07	12,345.67	D	cBBBB	Supplier B	1500	3730
Outlier type	Local						
Transaction type	Increase of debt to supplier						
Properties	In relation to a sub-selection of the entire transactional data set this synthetic outlier has an amount (*rgl_amt*) that relatively large in comparison to average of transactions related to customer or supplier (*cust_SupId*). Also the amount of the transaction is relatively large in comparison to the average of transactions posted on the 3730 ledger account (*gb_accId*).						

Table F.1: Local synthetic outlier, journal entry with high amount in sub-selection

tran_Nr	**rgl_trDt**	**rgl_amt**	**rgl_amntTp**	**cust_SupId**	**cust_SupName**	**dgb_jrnId**	**gb_accId**
19732	2017-05-03	50,573.54	D	cDDDD	Customer D	1300	1300
19732	2017-05-03	8,777.23	C	cDDDD	Customer D	1300	1511
19732	2017-05-03	41,796.31	C	cDDDD	Customer D	1300	8110
Outlier type	Local						
Transaction type	Sale of products						
Properties	In relation to a sub-selection of the entire transactional data set this synthetic outlier has an amount (*rgl_amt*) that relatively large in comparison to average of transactions related to customer or supplier (*cust_SupId*). Also the amount of the transaction is relatively large in comparison to the average of transactions posted on the 1511 ledger account (*gb_accId*).						

Table F.2: Local synthetic outlier, journal entry with high amount in sub-selection

tran_Nr	rgl_trDt	rgl_amt	rgl_amntTp	cust_SupId	cust_SupName	dgb_jrnId	gb_accId
21515	2019-08-03	567.32	C			2200	2200
21515	2019-08-03	526.08	D	cXXXX	Supplier X	2200	1500
21515	2019-08-03	41.24	D	cXXXX	Supplier X	2200	1500
Outlier type	Global						
Transaction type	Payment to supplier						
Properties	In relation to the transactional data set this synthetic outlier has a transaction date (*rgl_trDt*) after the financial year. Generally a financial statement audit is executed on a data set from a single previous financial year, transactions with a transaction date in a different year would be considered very anomalous.						

Table F.3: Global synthetic outlier, journal entry with date after financial year

tran_Nr	rgl_trDt	rgl_amt	rgl_amntTp	cust_SupId	cust_SupName	dgb_jrnId	gb_accId
21533	2019-08-07	321.26	D			2205	2205
21533	2019-08-07	331.2	C	dYYYY	Customer Y	2205	1300
21533	2019-08-07	9.94	D			2205	2205
Outlier type	Global						
Transaction type	Payment from customer						
Properties	In relation to the transactional data set this synthetic outlier has a transaction date (*rgl_trDt*) after the financial year. Generally a financial statement audit is executed on a data set from a previous financial year, transactions with a transaction date in a different year would be considered very anomalous.						

Table F.4: Global synthetic outlier, journal entry with date after financial year

tran_Nr	rgl_trDt	rgl_amt	rgl_amntTp	cust_SupId	cust_SupName	dgb_jrnId	gb_accId
18299	2016-02-16	95.75	C			2200	2200
18299	2016-02-16	95.75	D	cZZZZ	Supplier Z	2200	1500
Outlier type	Global						
Transaction type	Payment to supplier						
Properties	In relation to the transactional data set this synthetic outlier has a transaction date (*rgl_trDt*) before the financial year. Generally a financial statement audit is executed on a data set from a previous financial year, transactions with a transaction date in a different year would be considered very anomalous.						

Table F.5: Global synthetic outlier, journal entry with date before financial year

tran_Nr	rgl_trDt	rgl_amt	rgl_amntTp	cust_SupId	cust_SupName	dgb_jrnId	gb_accId
19857	2017-05-09	1,000,310.66	D			2200	2200
19857	2017-05-09	1,000,310.66	C	dAAAA	Customer A	2200	1300
Outlier type	Global						
Transaction type	Payment from customer						
Properties	In relation to the transactional data set this synthetic outlier has an amount (*rgl_amt*) that is one of the extreme large values in the entire set. This extreme amount in this data set is to be considered very anomalous.						

Table F.6: Global synthetic outlier, journal entry with extreme amount

tran_Nr	rgl_trDt	rgl_amt	rgl_amntTp	cust_SupId	cust_SupName	dgb_jrnld	gb_accld
20991	2017-04-07	1,001,869.45	D	cCCCC	Customer C	1300	1300
20991	2017-04-07	10,324.45	C	cCCCC	Customer C	1300	1511
20991	2017-07-07	1,001,545	C	cCCCC	Customer C	1300	8110
Outlier type	Global						
Transaction type	Sale of products						
Properties	In relation to the transactional data set this synthetic outlier has an amount (*rgl_amt*) that is one of the extreme large values in the entire set. This extreme amount in this data set is to be considered very anomalous. Alongside of the extreme amount the journal entry is not in balance which would considered very rare. The total debit amount (*rgl_amntTp* == D) should be equal to the total credit amount in a journal entry (*rgl_amntTp* == C).						

Table F.7: Global synthetic outlier, unbalanced journal entry with extreme amount

Appendix G

Microsoft Power BI implementation

Figure G.1 visualizes the outlier scores for transactions based on the Isolation Forests algorithm as currently implemented in one of the reports used during audits at de Jong & Laan Accountants. In this example the threshold is set at 0.6, therefore transactions scoring higher than the threshold are labeled with a red flag. Auditors are able to analyze these transactions in an efficient way, possibly finding transactions that could be of interest that they would've missed without these red flags.

Figure G.1: Power BI implementation of Isolation Forest algorithm