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

de Wit, R.C.

Award date:
2016

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
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.

by
Rogier de Wit

BSc Industrial Engineering – 2014
Student number: 0744077

In partial fulfillment of the requirements for the degree of

Master of Science
in Operations Management and Logistics

Supervisors:
Dr. Ir. R.M. Dijkman, TU/E
Dr. Rui Jorge de Almeida, TU/E
MSc. D. van de Wiel, Company supervisor
Subject headings:
Machine Learning, Anomaly Detection, Exception Detection, Data Analytics, Data Mining, Financial Audit, Transaction Level Data.
Abstract

This study examines the application of anomaly detection in the audit domain. Future audit practice ideally uses tools that examine 100% of clients transactions, identifying anomalies that guide auditors to the places that need additional test work. Machine learning algorithms have the potential to add flexibility and adaptability to current data examination methods, which mainly consist of specific rule based queries. The aim of this study is to show how anomaly detection algorithms can be applied in different audit related applications to automatically detect exceptions in datasets. By this investigation is shown how these algorithms could add value to the current audit and advisory practice. Anomaly detection algorithms are unsupervised machine learning algorithms, designed to identify unexpected items or events in datasets. In this thesis, the k-NN, uCBLOF and one-class SVM anomaly detection algorithms are applied in different audit related tests. In addition to tests on synthetic data, different datasets are used that consist of transaction details from companies to show the real life applicability of the algorithms. An extension to the Rapidminer data mining tool is used to do the analysis on the data and make the anomaly detection models. Models in this tool make anomaly detection available to general auditors. The k-NN algorithm performs best in the applications in this study. The algorithm is able to detect features with extreme values, infrequent occurring categories and infrequent occurring category combinations. This is shown to be applicable in different audit related tests and could be adopted in advisory tools.
Management summary

Background

Financial audit firms can nowadays benefit from access to huge amounts of client data. The availability of transaction level data has high potential. Assessment of this data could lead to the detection of exceptions like reporting errors, process deviations and clients risk areas that may not be found by using aggregated data in financial reports or samples of client data. In addition, new regulations induce an increased focus on analyzing transaction level data. With a focus on the future, the vice chair of audit at a public accounting firm states the following (Liddy, 2014, p.2):

“In the future, using high powered analytics, auditors will have the capacity to examine 100 percent of a client’s transactions. We will be able to sort, filter, and analyze tens of thousands of transactions to identify anomalies, making it easier to focus in on areas of potential concern and drill down on those items that may have the highest risks.”

Purpose of the study

The currently most used data examination method is extraction and query with specific hard coded rules. Machine learning algorithms have the potential to improve audit test by making them more flexible and adaptable to input data. The main research question addressed in this study is therefore formulated as follows:

How can machine learning be applied to detect exceptions, interesting to the auditor, in transaction level audit data?

The purpose of this study is to investigate different applications (in the audit environment) in which exceptions can be detected automatically by making use of anomaly detection algorithms to show how this type of algorithms could add value in the current audit and advisory practice.

Anomaly detection algorithms

In machine learning, the detection of not-normal data instances has always been of great interest. Anomaly detection algorithms are now used in many application domains and often enhance traditional rule-based detection systems. An anomalous transaction is one that is rare and deviates from the norm with respect to the value of his features. Auditors are interested in these transactions if the features (or combination of features) represent interesting aspect of a transaction.

Four main characteristics are considered for the selection of an anomaly detection algorithm: label availability, nature of the input data, anomaly type and the type of output. Unsupervised anomaly detection algorithms are used. The use of unsupervised algorithms is realistic due to the unavailability of labels that indicate if transactions are anomalous or normal in practice. An anomaly score is assigned to each transaction which makes it possible for the end user to adjust the threshold for what is considered anomalous in each situation and provides an indication of the anomalousness of the values of a transaction. The three selected and tested anomaly detection algorithms are the nearest neighbor based k-NN algorithm, the cluster based uCBOF algorithm and the support vector based one-class SVM algorithm. These algorithms are selected based on practical applications in the past and attention in the scientific community. Since the algorithms are available in the easy to use tool Rapidminer they could be used by general auditors to make them less dependent on IT auditors.

The algorithms use distance based measures to find anomalies. This has added value compared to usual statistical outlier detection techniques since they do not make an assumption about the distribution of the underlying data, multiple features can be assessed in one test and both categorical and continuous variables can be used. The exceptions are detected because they are rare in the dataset and deviate from the norm with respect to the value of their features.
Applications

In 5 applications is shown how anomaly detection can be applied. By showing multiple applications is aimed to provide a general indication of the capabilities and limitations of the anomaly detection algorithms in the audit domain and provide incentives to adopt the algorithms in other situations.

As a first application, anomaly detection is applied in a model to automatically detect sales transactions with anomalous values in different features. Transactions with extreme and rare values in their features are detected. Examples are a very high revenue or a long time between posting and the original document date. In addition, infrequent occurring categories are detected. This is for example an entry user that posts transactions very infrequent or an account number that occurs only once. Transactions that have multiple features with an extreme value or a rare category are highlighted. For example if there exists a transaction that has an extreme revenue, entered at night by an infrequent user.

Anomaly detection is also useful to automatically detect contextual anomalies. A contextual anomaly is a transaction that is only anomalous in a specific context but not otherwise. These values will not be found when using outlier detection in one feature only.

Anomaly detection is applied to detect invoices that have a long period of outstanding payment relative to the normal payment period for a specific supplier. In addition, the algorithms are applied to detect a payment that is made to the wrong supplier because two bank account numbers are switched. These applications are based on synthetic data.

Finally, two applications with real life data are tested. The first model is able to detect debit postings to accounts that are normally only credited and vice versa. The second model detects deviating tax codes in clients purchase transactions. A tax code can occur very frequent but rarely for a specific supplier. The k-NN anomaly detection algorithm is able to detect the transactions with a deviating tax code without any specific programming.

Conclusions

The k-nearest neighbor algorithm is in general the best performing algorithm. It has a true positive rate of 94% in the first model, a limited sensitivity to the algorithm parameter and the anomaly score provides a clear distinction between anomalous transactions and normal ones. Despite theory predicts other results, the computation time was in general the lowest for the k-NN algorithm (e.g. some seconds when analyzing a dataset of 40,000 records).

The combination of categorical and continuous variables in one analysis does not always provide the expected results due to the mixed distance measures. The distance based algorithms assign a ‘distance’ to the difference between two categories. Combining this with the distance between the points in a continuous variable is from a mathematical point of view not straightforward. Normalization of the features is of major importance if multiple features are combined in one test.

Since some big outliers in the data influence the results very much, data transformation from a continuous variable to a categorical variable can be used. This makes the results more robust. The k-NN algorithm provides good results in finding infrequent occurring combinations when analyzing multiple categorical features together.

Anomaly detection only considers intrinsic information of the dataset which can result in many false positive results in practice. Data points that are rare do not have to be interesting exceptions but can just result from normal course of business. Limiting false positives in the output can be done by combining anomaly detection with programmed rules. As an alternative, supervised algorithms have the ability to learn from user input and are able to learn which combination of parameter values are considered anomalous.
Recommendations

The k-NN anomaly detection algorithm is the simplest but most effective algorithm based on this study. The algorithm is inherently unsupervised, it has the most intuitive criteria for the detection of outliers and is therefore recommended to use above the uCBLOF and the one-class SVM algorithm. The algorithm also was the fastest of the three algorithms. It should however be noted that when analyzing really large numbers of data (i.e. millions) the algorithm can perform less due to exponential computational complexity.

An additional model in Rapidminer for auditors to quickly identify deviating transactions, risk areas and process deviations can be adopted to analyze for example sales or purchase transactions of clients and will contribute to a more data driven audit. Transactions with extreme and rare occurring values in the included features will be identified. It is important to provide the right input data. Normalization of all data such that different features can be compared. Data transformation steps may be required for the algorithm to work well (for example transforming the feature date to an integer value or aggregate a feature to find the frequency of occurring). It is also important to only include the parameters of interest because irrelevant features will lead to noisy results instead of adding to a possible pattern. An anomaly detection model should be used as additional analysis and not as replacement of other methods. When for example errors occur in large quantities (e.g. frequent deviation from a process or frequent occurring deviating payments) the technique will not work.

The k-NN algorithm is especially good in the detection of infrequent (combinations of) categories. For example, when a bank account number is only once used in combination with a specific supplier name, it can be detected with an anomaly detection algorithm. In this study is shown how exceptional tax codes can be detected when they deviate from the tax codes that are used in general for the transactions with a specific supplier. There can be further investigated how anomaly detection can be adopted in audit/advisory tools to make them more flexible and adaptable to input data.

In this study is investigated how anomaly detection can be applied to determine whether it has potential to improve current rule based methods. Specific queries always generate the results that follow from the code. By contrast, the anomaly detection algorithms are standard algorithms which require the data to be supplied in such a way that the rare and deviating data points are interesting. Only selecting relevant features and data transformation steps are required for optimal performance of the algorithms. Anomaly detection is in this way applicable in a lot of different test and analysis for financial audit.
Preface

In the spring of 2016 I entered the building I was always cycling along during my time at Eindhoven University of Technology. I remembered the nice view on Eindhoven at the 9th floor from a lunch lecture with my study association. With the accounting knowledge I gained as a treasurer during my board year as backup, I started my master thesis project at the audit and advisory firm.

Now, six months later, I am almost finished with my thesis and look back on an inspiring time in which I learned a lot (beside the subject of my study) about how a professional firm is organized and how they add value to their clients every day. The motivated people around you and the stimulating work environment at the company helped a lot to keep the drive needed to successfully complete the project. It was also nice to experience that graduates are part of the team from the first moment they come in. Some events like the ‘innovation conference’ were also inspiring and nice to participate in.

Since the firm always wants to stay ahead of their competitors and bring clients state of the art solutions, my project was quite an exploration of new possibilities. Machine learning is one of the big hypes with high expectations to change business operations, but how the techniques exactly can be applied and how they would add value is in many industries not clear yet. With a lack of other research in this area, the application of machine learning to detect deviating transactions was a challenge. Also since I only followed one course in Business Intelligence and did not have much audit knowledge, there was a lot to learn during my project. After all, I am happy to share the results with you via this report.

This study would not have been possible without the help of other people. Therefore, I would like to thank some people here for their support. I would like to thank Remco Dijkman for his supervision during the whole project. In addition, I would like to thank Dennis and Elham for their advice during my project from the company side. Also thanks to Joris for the discussions about our projects and of course for the tips to improve my table soccer skills. Overall, thanks to all people at the company with whom I talked about my project or helped me along the way. Finally, I would thank my mom and brother for the support during my studies and the relaxing weekends at home.

I hope that you will enjoy reading my report and hopefully the gained knowledge will inspire to be adopted in real business operations.

Rogier de Wit
November 18, 2016
# Table of Contents

List of Figures xiii
List of Tables xv
List of Abbreviations xvi

Chapter 1: Introduction & research design

1.1 Introduction ................................................................................................................................ 1
1.2 Research domain.......................................................................................................................... 2
1.3 Research goal .............................................................................................................................. 3
1.4 Research design .......................................................................................................................... 4
1.5 Contributions and work developed ............................................................................................ 5
1.6 Thesis structure ........................................................................................................................... 6

Chapter 2: Current data analysis in audit

2.1 The purpose of auditing .............................................................................................................. 7
2.2 Description of risks related to exceptions in sales transactions ................................................. 7
2.3 Change in audit and future opportunities .................................................................................. 9
2.4 Current data examination approach ........................................................................................... 9
2.5 Current audit tools .................................................................................................................... 12
2.6 Capabilities of the auditor ......................................................................................................... 12
2.7 Conclusions ............................................................................................................................... 12

Chapter 3: Anomaly detection algorithms and the link to transaction level audit data

3.1 Introduction and definitions of anomaly detection .................................................................. 13
3.2 Key components associated with anomaly detection techniques ............................................. 15
    3.2.1 Nature of the input data ................................................................................................... 15
    3.2.2 Type of anomaly ............................................................................................................ 16
    3.2.3 Data labels ..................................................................................................................... 17
    3.2.4 Output of anomaly detection algorithm ........................................................................... 18
    3.2.5 Conclusions ..................................................................................................................... 18
3.3 Selection of the anomaly detection algorithms ........................................................................ 19
    3.3.1 Taxonomy of unsupervised anomaly detection ............................................................... 19
    3.3.2 Selection of anomaly detection algorithms .................................................................... 19
    3.3.3 Discussion of selected algorithms .................................................................................. 20

Chapter 4: Point anomaly detection model

4.1 Problem description & unit of analysis ..................................................................................... 22
4.2 Data description ....................................................................................................................... 23
4.2.1 Feature selection criteria .................................................................................................. 23
4.2.2 Defining exceptions to be found by the anomaly detection algorithms ......................... 24
4.3 Modeling in Rapidminer ............................................................................................................ 26
4.4 Test design ................................................................................................................................ 26
4.5 Results ....................................................................................................................................... 26
4.6 Evaluation ................................................................................................................................. 27
4.7 Conclusion ................................................................................................................................. 29
4.8 Deployment of the model ......................................................................................................... 30
4.9 Extension of the model ............................................................................................................. 31

Chapter 5: Contextual anomaly detection

5.1 Introduction to contextual anomaly detection in audit ........................................................... 32
5.2 Anomaly detection model with synthetic data ....................................................................... 33
  5.2.1 Two dimensional anomaly detection problem ................................................................. 34
  5.2.2 Three dimensional anomaly detection problem ............................................................... 35
  5.2.3 Conclusion ......................................................................................................................... 37
5.3 Anomaly detection model with real world data ....................................................................... 38
  5.3.1 Detection of debit (credit) postings to typical credit (debit) accounts............................. 38
  5.3.2 Detection of tax code anomalies in purchase transactions .............................................. 41
5.4 Include feedback to improve an anomaly detection algorithm output and create learning effect.................................................................................................................................45

Chapter 6: Conclusions, limitations & further research

6.1 Conclusions ............................................................................................................................... 47
6.2 Limitations and further research .............................................................................................. 49

Appendices

Appendix A: Gartner Hype Cycle of Technology 2016 ................................................................. 54
Appendix B: Summary of unstructured interviews conducted to find appropriate transaction type for the application of anomaly detection ......................................................................... 55
Appendix C: Description of selected features of the sales transaction data .................................. 56
Appendix D: Summary semi-structured interviews with audit managers ...................................... 58
Appendix E: Descriptive statistics and plots of the values of the eight features used in the anomaly detection model for sales transactions ........................................................................ 63
Appendix F: How to set the parameter of the anomaly detection algorithm .................................. 64
Appendix G: Data labeling procedure ............................................................................................ 66
Appendix H: Anomaly detection performance per feature ............................................................ 67
Appendix I: Top 10 outliers per algorithm (with optimal parameter setting) ................................. 68
Appendix J: Results two dimensional anomaly detection algorithm with synthetic data ............. 69
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Developed research design</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Report structure</td>
<td>6</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Data examination process in audit (based on Gray &amp; Debreceny, 2014)</td>
<td>10</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Relationship of data examination techniques (Gray &amp; Debreceny, 2014)</td>
<td>11</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Example of anomalies in a two-dimensional dataset (based on Chandola, Banerjee &amp; Kumar, 2009)</td>
<td>14</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Key components associated with an anomaly detection technique (based on Chandola et al., 2009)</td>
<td>15</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Different anomaly detection modes depending on the availability of labels in the dataset (based on Goldstein &amp; Uchida, 2016)</td>
<td>17</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Taxonomy of unsupervised anomaly detection algorithms (based on Goldstein &amp; Uchida, 2016)</td>
<td>19</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Values in feature 'doc date-entry date'</td>
<td>25</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Values in feature 'profit'</td>
<td>25</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Values in feature 'cost'</td>
<td>26</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Values in feature 'entry time'</td>
<td>26</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Values in feature 'delivery surplus'</td>
<td>26</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Values in feature 'entry user'</td>
<td>26</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Values in feature 'profit percentage'</td>
<td>26</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Values in feature 'revenue'</td>
<td>26</td>
</tr>
<tr>
<td>Figure 17</td>
<td>ROC curves of different anomaly detection algorithms</td>
<td>27</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Visualization of Rapidminer output (cost vs. entry user)</td>
<td>28</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Overview anomaly detection process</td>
<td>30</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Theoretical example two dimensional anomalies</td>
<td>32</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Practical example two dimensional data</td>
<td>32</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Synthetic example of contextual anomaly detection</td>
<td>33</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Results one-class SVM anomaly detection algorithm (color represents anomaly score)</td>
<td>34</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Visualization results k-NN algorithm</td>
<td>36</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Visualization debit/Credit anomalies</td>
<td>39</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Anomaly detection process</td>
<td>40</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Visualization tax anomalies</td>
<td>42</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Visualization tax anomalies per value</td>
<td>44</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Tax anomalies, date (integer) vs. tax code</td>
<td>44</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Tax anomalies, date (aggregated per month) vs. tax code</td>
<td>44</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Anomaly detection process 'tax anomalies'</td>
<td>45</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Original output tax code anomaly model</td>
<td>46</td>
</tr>
<tr>
<td>Figure 33</td>
<td>Output tax code anomaly model with rules</td>
<td>46</td>
</tr>
<tr>
<td>Figure 34</td>
<td>Gartner Hype Cycle of Technology</td>
<td>54</td>
</tr>
<tr>
<td>Figure 35</td>
<td>Visualization of the elbow rule (based on Leskovec &amp; Rajaraman, n.d.)</td>
<td>64</td>
</tr>
<tr>
<td>Figure 36</td>
<td>Average within cluster distance</td>
<td>64</td>
</tr>
<tr>
<td>Figure 37</td>
<td>Visualization k-NN algorithm (late invoice payments)</td>
<td>69</td>
</tr>
<tr>
<td>Figure 38</td>
<td>Visualization one-class SVM algorithm (late invoice payments)</td>
<td>69</td>
</tr>
<tr>
<td>Figure 39</td>
<td>Visualization uCBLOF algorithm (late invoice payments)</td>
<td>69</td>
</tr>
<tr>
<td>Figure 40</td>
<td>Output k-NN algorithm (date not normalized)</td>
<td>70</td>
</tr>
<tr>
<td>Figure 41</td>
<td>Output uCBLOF algorithm (date not normalized)</td>
<td>70</td>
</tr>
</tbody>
</table>
Figure 42: Output one-class SVM (date not normalized) ................................................................. 70
Figure 43: Visualization k-NN algorithm (switched account numbers) ........................................... 71
Figure 44: Visualization one-class SVM algorithm (switched account numbers) ............................. 71
Figure 45: Visualization uCBLOF (switched account numbers) ....................................................... 71
Figure 46: Usual type of postings to account .................................................................................... 72
List of Tables

Table 1: Technological improvements that affected audit (based on Kuenkaikaew, 2013) .............. 2
Table 2: Big data and the impact on audit approach ................................................................. 9
Table 3: Selected/Generated features for anomaly detection model .......................................... 23
Table 4: Definition of normal and exceptional values for sales transaction features ............ 25
Table 5: Test design .................................................................................................................. 26
Table 6: Performance of different anomaly detection algorithms ............................................. 26
Table 7: Overview percentages of transactions assigned to different categories .................... 29
Table 8: Top 5 output uCBLOF anomaly detection algorithm ................................................. 30
Table 9: Results for different weight factors of the feature delivery surplus ......................... 31
Table 10: Synthetic data, days payment outstanding per company ......................................... 34
Table 11: Anomaly score of different algorithms ..................................................................... 34
Table 12: Overview synthetic data with features 'date', 'supplier' and 'amount' ....................... 35
Table 13: Output different anomaly detection algorithms ....................................................... 36
Table 14: Example journal entry ............................................................................................. 38
Table 15: Output anomaly detection model ............................................................................. 39
Table 16: Tax code anomaly detection output ......................................................................... 42
Table 17: Comparison of main characteristics anomaly detection vs. supervised learning ....... 45
Table 18: Selected features from dataset ............................................................................... 56
Table 19: Descriptive statistics dataset ................................................................................... 63
Table 20: Example of transaction labels used for evaluation ................................................. 66
Table 21: True positive rate per feature k-NN algorithm ......................................................... 67
Table 22: True positive rate per feature one-class SVM ......................................................... 67
Table 23: True positive rate per feature uCBLOF ................................................................. 67
Table 24: Top 10 results k-NN algorithm ............................................................................... 68
Table 25: Top 10 results one-class SVM ............................................................................... 68
Table 26: Top 10 results uCBLOF ......................................................................................... 68
Table 27: Output k-NN algorithm (late invoice payment) ......................................................... 69
Table 28: Output one-class SVM algorithm (late invoice payments) ....................................... 69
Table 29: Output uCBLOF algorithm (late invoice payments) ............................................... 69
Table 30: Validation point anomaly detection model output .................................................... 75
Table 31: Validation point anomaly detection model output (2) ............................................ 76
List of Abbreviations

AIS  Accounting Information System
ASB  Auditing Standards Board
CAAT Computer Assisted Audit Tool
CMGOS  Clustering-Based Multivariate Gaussian Outlier Score
ERP-system  Enterprise Resource Planning System
GAAP  Generally Accepted Accounting Principles
HBOS  Histogram Based Outlier Score
IAASB  International Auditing and Assurance Standards Board
IS  Information System
IT auditor  Information Technology Auditor
k-NN  k-Nearest Neighbor
PCA  Principal Component Analysis
SAS  Statement of Auditing Standards
SOX  Sarbanes Oxley Act
SQL  Structured Query Language
SVM  Support Vector Machine
uCBLOF  unweighted Cluster Based Local Outlier Factor
VAT  Value Added Tax
1. Introduction & research design

The most recent Gartner Hype Cycle for Emerging Technologies classifies machine learning as the technology which is expected to have the greatest impact on organization’s strategic planning in the next two to five years (Appendix A). The combination of smart data discovery and endless amounts of data will be the most disruptive technology over the next 10 years (Gartner, 2016). This thesis investigates if, in line with these trends, anomaly detection algorithms can add value for auditors.

In this chapter the topic of this study is introduced. The chapter starts with a general introduction. Afterwards is discussed which research gap is addressed by this study. This leads to the central research question. Subsequently, contributions of the study to both literature and the business are set out. This chapter ends with a description of the research design and an overview of the structure of this thesis.

1.1 Introduction

Every company participates in transactions. For example, when they pay their suppliers or when they receive money when an invoice is paid. As you can imagine, the number of transactions in modern businesses can reach up to thousands or even millions a year. All transactions are recorded in ERP systems which provide the richest source of information in a company (Werner & Gehrke, 2015). This study focusses on this transaction level type of data. During the processing and reporting of the transactions it is possible that errors occur, either by honest mistake or due to fraudulent activity. An unintentional error could be a clerk that switches two bank account numbers causing an amount being transferred to a wrong supplier. An intentional misstatement is executed by e.g. manipulating entries related to sales, profit, expenses and losses or misappropriation of taxes resulting in financial figures of a company better than it actually is (Sharma & Panigrahi, 2012).

Because these errors could occur, an external audit company investigates the financial reports and reporting process. In general, the task of an auditor is to “plan and perform the audit to obtain reasonable assurance about whether the financial statements are free from material misstatement, whether caused by error or fraud” (Bay et al., 2002). In addition, there is especially in the last years an increased focus of larger accounting firms on non-audit services (e.g. advisory). This trend increases the value of identifying exceptions that may not directly conflict with accounting rules but could indicate process deviations or other business problems (Current Trends in the Audit Industry, 2015). An example is a wrong tax code that is entered on a purchase transaction resulting in redundant VAT payments and unnecessary loss of money for a company.

Exceptions need to be found in transaction level data although this is not an easy task (Chandola, Banerjee & Kumar, 2009). Current methods include search with specific queries to identify pre-defined exceptions. In the machine learning field, the detection of not-normal data instances within datasets is commonly known as anomaly detection (Goldstein & Uchida, 2016). These unsupervised algorithms learn based on the intrinsic information of the dataset and are therefore very flexible (Amer, Goldstein & Abdennadher, 2013). They are applicable when the nature of the anomaly is constantly changing and not known in advance.

With the use of anomaly detection algorithms, exceptions in transaction level data can be found automatically and guide auditors to transactions of interest. In this research different algorithms are tested to identify which anomaly detection algorithm performs best for the anomalies in the financial audit domain. Furthermore, multiple situations in which anomaly detection can add value are identified, tested and evaluated. In this way a first step is made in the application of unsupervised anomaly detection in the financial audit domain.
1.2 Research domain

In this paragraph, an overview of the literature in this research domain is provided to show how this research is linked to other literature in this research domain.

Two main focus areas can be distinguished in relation to data mining research within the financial audit domain (Kuenkaikaew, 2013). In the first category, there is focused on aggregated financial data. An example of aggregated data is the data in yearly or quarterly financial statements. Most of the work in the area of detecting accounting irregularities is based on aggregated data and this research area has been proven to be a well-researched area (Sharma & Panigrahi, 2012). Yet, there have been substantial technological improvements that have affected auditing (Table 1).

<table>
<thead>
<tr>
<th>Factors</th>
<th>Past</th>
<th>Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data availability</td>
<td>- Scarce</td>
<td>- Abundant</td>
</tr>
<tr>
<td></td>
<td>- Most data in paper format</td>
<td>- Most data is digital</td>
</tr>
<tr>
<td>Access to data</td>
<td>- Limited</td>
<td>- Mostly unlimited</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Audit data warehouse</td>
</tr>
<tr>
<td>Data storage</td>
<td>- Limited capacity</td>
<td>- Large (i.e. terabytes)</td>
</tr>
<tr>
<td></td>
<td>- Expensive</td>
<td>- Cheaper</td>
</tr>
</tbody>
</table>

Table 1: Technological improvements that affected audit (based on Kuenkaikaew, 2013)

These technological improvements lead towards the second and emerging category of data mining research in the financial audit domain. In this category is focused on disaggregated data.

Disaggregated data consists of the transaction level data in ERP systems. According to Werner & Gehrke (2015) this rich source of information remains mostly untouched by current audit tools. The pure fact that more data is available does not mean it is interesting by itself. However, there are many studies that advocate research in this direction. Bay et al. (2006) state that effective fraud detection analytics should operate at the level of detailed transactions. Not only are these potentially more accurate, but an important side benefit is the possibility to detect transactions that may not be fraudulent but indicate errors or control deficiencies that are also of interest to auditors. Cao, Chychala & Stewart (2015) hypothesizes that big data analytics can improve the efficiency and effectiveness of financial statement audits. Jans, Lybaert & Vanhoof (2007) conclude that research in unsupervised learning in the audit domain is due for catching up. Despite all encouragement, most papers have not studied data at a transactional level yet (Kuenkaikaew, 2013). Debreceny & Gray (2010) explain this by the fact that researchers lack access to transaction level datasets.

This study is conducted at a public accounting firm which has access to the data that is needed to perform research on disaggregated, transaction level data. There can be concluded that there is a lack of existing research on the detection of exceptions in transaction level data while promising results are expected. Therefore, this gap will be addressed in this report.
1.3 Research goal

This study aims to identify ways to make the audit of a company both more efficient and effective. Effectiveness is about increasing the probability to find errors. Efficiency involves increased automation and technology to decrease the time and effort to perform an audit (Chan & Vasarhelyi, 2011). Two research fields are combined to make this step forward possible.

Research field 1: Data driven audit with transaction level data

The audit of transaction level data is concerned with finding exceptions in three categories (Bay et al., 2002). The first category contains unusual transactions that an auditor would consider worth investigating, even if they are found to be neither fraud nor error. The second category consists of unintentional errors, such as clerical mistakes when entering data. The last category contains exceptions due to fraud.

A data driven approach could guide the auditor to the exceptional transactions and processes that contain deviations. The increased focus of public accounting firms on non-audit services is also one of the drivers for this focus. So this research is mainly focused on the first two categories. However, since over one third of the accounting fraud is discovered by chance (Jans et al., 2007) the data driven audit approach could also add value in finding (indicators of) fraud.

Research field 2: Anomaly detection

In machine learning, the detection of not-normal instances within datasets has always been of great interest (Goldstein & Uchida, 2016). Within the machine learning field, anomaly detection algorithms are now used in many application domains and often enhance traditional rule-based detection systems. These aspects could improve the current data examination methods in terms of flexibility and adaptability. Anomaly detection algorithms are standard algorithms that can be applied in different situations and adapt on input data rather than follow strict queries.

Public accounting firms perform audit engagements at various different companies. They differ in size, industry and type of risks. The different data analytical procedures should therefore also adjust to the situation and the data. To give a simple example, an exceptional high value transaction has a very different absolute value in a small company compared to a large multinational. Due to recent regulatory changes (Kantoorroulatie, 2016) the public accountants are also forced to limit their work for one company to a period of ten years which leads to more changes in clients. More flexibility and adaptability in audit tests is overall desirable.

Central research question

Combining both research fields results in the main research question of this study which is formulated in the following way:

*How can machine learning be applied to detect exceptions, interesting to the auditor, in transaction level audit data?*
1.4 Research design

Since the research field is not well developed yet, the nature of this study is exploratory. Blumberg, Cooper & Schindler (2011) define an exploratory study as the process of gathering background information on the topic to expand understanding of the research dilemma and discover future research tasks.

The methodology of Hevner et al. (2004) is chosen because it is well known in information systems research and it provides guidelines for research in fields that are not well developed yet. In addition, it helps to balance between rigor and relevance, where rigor refers to the development of theory and relevance refers to the applicability of the research (Hevner et al., 2004). Based on the methodology of Hevner et al. (2004), the research question is divided in four parts. The four parts are visualized in Figure 1.

In the first part, the current business situation related to the audit of transaction level data is investigated which leads to the identification of the ‘business needs and opportunities’. The data examination process is investigated and current audit tools and capabilities of (IT) auditors are discussed to get an understanding of the current audit procedures.

The second part of this research consists of the identification of the ‘knowledge base’. Literature is examined to get a throughout understanding of the applicable machine learning techniques. Characteristics of anomaly detection are examined and the most appropriate algorithms are selected based on the key components of anomaly detection in transaction level data in the audit environment.

With the knowledge about anomaly detection, the appropriate algorithms and background knowledge about on the audit environment, an anomaly detection model is built to help auditors in the guidance to deviating sales transactions. The model is tested with a case study on real client data and afterwards evaluated with auditors. For deployment in practice is investigated in which stages of the audit process and on which other sources of data the model could be used.

The fourth part of this study follows from the evaluation of the model in part three and is focused on the detection of contextual anomalies in audit related tests. To show and test the capabilities of the algorithms is started with analysis based on synthetic data. Afterwards, real life applications will be identified. Finally, there will be investigated if it is possible to provide feedback (learning effect) to the unsupervised anomaly detection algorithm.

Each step corresponds respectively to chapter 2, 3, 4 and 5 in the report as shown in Figure 2.

Figure 1: Developed research design
1.5 Contributions and work developed

By answering the research question several contributions will be made to both literature and business. These will be discussed in this paragraph.

This thesis will show how the current data examination process in audit is executed. By doing this, the place where anomaly detection algorithms could add value in this process will be determined. Also several other business aspects will be discussed to provide a good understanding of the business needs.

The key components associated with the detection of exceptions will be determined. This will result in a good understanding of the problem characteristics specifically linked to exceptions in the audit environment.

Within machine learning many different algorithms exist, each with their own strong points and drawbacks. Three different algorithms that are suitable for anomaly detection in the audit environment are selected. This will be done based on the problem characteristics in the audit environment defined earlier. These three algorithms are applied in the models that are made. In this way can be concluded which algorithm works best for the detection of exceptions in transaction level data in audit. This will be advantageous when an algorithm should be selected for practical use in the business environment.

In this thesis, the selected anomaly detection algorithms will be tested on different kinds of datasets. Each dataset represents a separate audit related problem type. In this way are five unique applications of anomaly detection in audit identified and is shown how the algorithms perform on these problems. The knowledge about what is possible with anomaly detection in different situations can be used as a guideline for further research on anomaly detection in the audit environment.

By evaluating the models with different stakeholders will be identified how the algorithms can be used in practice. The audit of a company consists of different stages and in each stage the anomaly detection algorithms could add value in a different way. Additional applications of anomaly detection in audit will also be identified during the interviews. In this way will be shown that anomaly detection is very broadly applicable in the audit domain.

By making use of a tool that is easy to use and understand, auditors will be able to adopt the more sophisticated data analytical procedures in their day to day work. This would contribute directly to the effectiveness and efficiency of the audit tests. In addition, showing what the possibilities of anomaly detection are in the audit domain provides incentives for the adoption of anomaly detection algorithms in audit and advisory tools.
1.6 Thesis structure

In the previous paragraphs the subject of the research is introduced, the research gap and research goals are discussed and the research design is explained. In this paragraph will be shown how the remainder of this report is structured. Figure 2 provides an overview of the structure of this report.

In chapter 1, all relevant aspects related to the research question are discussed. The actual research starts in chapter 2 with the investigation of the audit environment. The current data examination process in audit will be discussed and there will be investigated how machine learning algorithms could add value. In chapter 3, literature about anomaly detection algorithms is investigated and linked to the audit environment. Different algorithms will be selected that are suited to the problem of detecting exceptions in transaction level audit data. After investigation of the business needs and selection of appropriate machine learning algorithms an anomaly detection model is proposed in chapter 4. In this model the performance of the different algorithms is compared. By evaluating the model with the end user, different auditors, is investigated how the model performs and how it can add value in practice. In chapter 5, contextual anomalies are investigated. Multiple situations are identified in which is shown how anomaly detection algorithms could be applied to detect contextual anomalies in transaction level audit data. Finally, in chapter 6, the research question is answered, limitations of this study are discussed and directions for further research are provided.

Figure 2: Report structure
Chapter 2: Current data analysis in audit

In this chapter the business environment of this study is investigated. Every company needs to be audited by an external audit firm to provide assurance that the financial results are reported correctly. Current information systems of a business consist of enormous amounts of data. This transaction level data is getting more and more attention in the audit process because it can guide auditors to the transactions of interest. Anomalous (sales) transactions can indicate high risk, an error or a control deficiency. The sophistication of the current data examination process can be improved by making more use of data mining. A more sophisticated test requires more advanced tools or algorithms to be used that are still easy to use for the auditor.

This chapter is structured as follows. There is started with a brief outline of the purpose of financial auditing in 2.1. In paragraph 2.2 is outlined why exceptions in sales transactions are interesting to be detected for use in the model in chapter 4. Paragraph 2.3 contains opportunities for unsupervised learning defined by domain experts. In 2.4, the current data examination process is set out. Finally the current use of audit tools and capabilities of the auditor are discussed shortly.

2.1 The purpose of auditing

A company’s need for the audit of their financial statements by an independent external auditor has been a cornerstone of confidence in the world’s financial system. A financial audit is an independent and objective evaluation of both the organization’s financial report and reporting process (Arens, Elder & Beasley, 2012). The primary purpose of the execution of a financial audit is to provide stakeholders reasonable assurance that the financial statements are accurate and complete. Regulators want to make sure the organization follows laws and present their financial figures accurately due to, for example, tax reasons. Investors should have the assurance they can rely on the presented figures for their investment. Within the company, managers and directors need the assurance all relevant controls are in place.

The audit is done in accordance with a framework of generally accepted accounting principles (GAAP) relevant to the country. In a response to significant financial statement frauds over the last decade, the regulatory environment has made a number of changes to these auditing standards. The announcement from the Auditing Standards Board (ASB) of SAS 99 and the Sarbanes-Oxley Act (SOX) introduced by the U.S. government were central events (Debreceny & Gray, 2010). The International Auditing and Assurance Standards Board (IAASB) adapted the same kind of standards in IAS 240. New statements in the standards include the responsibility of auditors to address potential fraud by direct assessment of journal entries (Argyrou, 2013). Therefore, the focus of auditors is more and more on direct assessment of transaction level data. Investigating new and innovative approaches to analyze the transaction level data is therefore of major importance for public accountant firms.

2.2 Description of risks related to exceptions in sales transactions

As described in the previous section, transaction level data should get more attention in the audit of a company. In chapter 4, an anomaly detection model is proposed to find anomalies in sales transaction data. The business risks related to exceptions in features of sales transactions are discussed in this paragraph.
The sales process of a company has to be audited to make sure all orders are recorded in the right way in the ERP systems. The auditors should have reasonable assurance the transactions are recorded correctly. Because the administration of sales transactions is a manual process, there is a certain risk that errors occur. These errors can occur in a variety of features and due to multiple reasons. It is therefore not aimed to provide an inclusive analysis of all possible risks related to sales transactions. It is rather aimed to identify features that could contain anomalous values that are interesting to the auditor. In this way could unsupervised machine learning (anomaly detection) be applied in chapter 4, to make current tests more efficient and effective.

**Profit & Loss Statement risks**

When a sales transaction is recorded in the ERP system, various accounts are affected. An order typically has cost associated with it to produce the product. These costs are referred to as ‘costs of goods sold’ and will appear on the corresponding cost of goods sold account on the profit & loss statement. An order will typically also include revenue, the amount paid by the customer, which will influence the sales account (via accounts receivable). If certain orders have very high cost or very high revenue compared to the usual orders, those specific orders will influence the corresponding accounts very much. It is therefore important for an auditor to identify and review especially these large and infrequent orders more throughout (Boynton & Johnson, 2006). This throughout analysis includes reviewing different original source documents like shipping documents, cash receipts and written correspondence between client and customer (Boynton & Johnson, 2006). Automatically detecting these orders would have added value.

**Risks of manual processing**

The administration of sales orders into the ERP system is a manual process. Because it is a manual process it is possible that various errors occur. The manual aspect also provides room for manipulation of the sales recordings. Johnstone et al. (2013) describes various incentives of management or employees to misstate sales recordings. For example, to meet a certain amount of sales in a period to get a bonus. It is therefore interesting to investigate if there are orders processed by users that do not regularly perform this action (e.g. exceptional entry users). It can also be interesting to see if there are orders processed on irregular times of the day (e.g. outside business hours). A final example of a risk associated with misstatement of sales is channel stuffing (Markham, 2015). Channel stuffing is a way to manipulate accounts and manage earnings by delivering more goods than needed. Usually there is a right of return which means the goods would be returned on a later date. Revenues are in this way only temporary inflated. In general, a rare number of transactions with deviating characteristics can indicate a deviation from a process which is also interesting to an auditor.

So there can be concluded that transactions with extreme values (very large or small compared to what is usual for a company) are interesting to the auditor. This is consistent with for example the research of Argyrou (2013) on extreme value theory. They focus specifically on the monetary value of transactions and define ‘suspicious’ transactions as having a large amount and low probability of occurring.

In the proposed anomaly detection model in chapter 4, multiple features can be analyzed together. The application of anomaly detection has added value in this context because ‘what is normal’ and ‘what is an anomalous value’ is different in every situation and every company. The model is therefore very flexible to use and does not rely on static programmed rules. An example of such a static rule can be: select all orders above €1.000.000,-. With the anomaly detection technique it will only highlight exceptional transactions that have both a high value and are rare.
2.3 Change in audit and future opportunities

The objective of research in information systems is to acquire knowledge and understanding that enables the development of technology based solutions that fulfil business needs (Hevner et al., 2004). The environment in which the information system research is performed consist of people, organizations and existing or planned technologies (Silver et al., 1995). In this environment arise opportunities that define business needs as they are perceived by people in the organization. The opportunity investigated in this study is the application of machine learning techniques in the audit domain. More specifically, how unsupervised machine learning can be applied on transactional data to find interesting exceptions for the auditor. This application is perceived to be a very promising opportunity in the audit sector. According to the Vice Chair of audit of one of the big four audit firms (Liddy, 2014, p.2):

“In the future, using high powered analytics, auditors will have the capacity to examine 100 percent of a client’s transactions. We will be able to sort, filter, and analyze tens of thousands of transactions to identify anomalies, making it easier to focus in on areas of potential concern and drill down on those items that may have the highest risks.”

In addition, Earley (2015) defined the current and potential future practice in audit by making use of data analysis, shown in Table 2.

Table 2: The impact of data on audit approach

<table>
<thead>
<tr>
<th>TYPE OF DATA</th>
<th>CURRENT PRACTICE</th>
<th>POTENTIAL FUTURE PRACTICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINANCIAL/TRANSACTIONAL DATA</td>
<td>Auditors collect and test a sample of transactions and use judgement on those areas that are difficult to test.</td>
<td>Tools can test 100% of transactions. Will identify anomalies in client-provided transaction data. This will guide additional test work, possibly uncovering fraudulent transactions. Judgement used in assessing next steps after anomalies are uncovered.</td>
</tr>
</tbody>
</table>

In this study is investigated how unsupervised machine learning can contribute to this goal. These opportunities are linked to the current data examination process in audit in the next paragraph. This is done to position and link this research within the current data analytics process in the audit domain. In this way is mapped how the current situation looks like and which business needs can be fulfilled by this research.

2.4 Current data examination approach

The common amount of records in modern accounting information systems of big companies can reach up to hundreds of thousands of transactions per accounting period (Debreceny & Gray, 2010). Manually addressing these for error and fraud detection is therefore infeasible (Lanza, Gilbert, & Lamoreaux, 2007). The total data examination process in the current audit is structured in Figure 3.
Figure 3: Data examination process in audit (based on Gray & Debreceny, 2014)
The audit starts with four processes to investigate which data should be investigated:

- Firms policies & procedures
  - The external auditor has their own standard policies and procedures for the planning of the audit.
- Walk through audit templates
  - Audit templates that are designed by the external audit firm for specific audit objectives.
- Audit team brainstorming
  - The audit team will discuss if there are specific risks for the a specific client.
- Individual auditor decisions
  - Based on experience of auditor or based on past audits at the company.

Afterwards the relevant databases and tables are selected and downloaded to a server of the external audit firm. From here on the data examination can be performed. Three groups can be distinguished within data examination based on the level of software sophistication and diagnostic power (Gray & Debreceny (2014)):

- Perform data extraction and query (e.g. find all transactions with an amount above 10.000).
- Perform data analysis (e.g. get descriptive statistics like minimum, maximum, means, medians, etc.)
- Perform data mining (e.g. discover patterns or models to find transactions that fall outside predicted ranges which the auditor should investigate (Gray & Debreceny, 2014)).

The conceptual relationship between data extraction, data analysis and data mining is shown in Figure 4.

As illustrated in Figure 4, data analysis and data mining overlap. The triangle in Figure 4 represents the relative frequency of use of the techniques by auditors. As shown, data mining is used very limited in the current accounting practice. The place of this research within the business environment is mapped by the green square in Figure 3. Unsupervised anomaly detection can add value as a data mining technique because it can pinpoint transactions that are exceptional and thereby increase the auditors productivity and effectiveness.
2.5 Current audit tools

The different data examination categories discussed in the previous paragraph require different tools. The data examination tools that are used in the audit process are in literature described by the term ‘Computer Assisted Audit Tools’ or briefly as CAAT’s (Werner & Gehrke, 2015). The tool that is most often used is Microsoft Excel (Curtis & Payne, 2014). Other CAAT’s are targeted specifically to the audit sector. Examples of these tools are ACL and IDEA which are used for data query and analysis and include a variety of options for descriptive statistics and a limited number of statistical techniques. They also include options to prepare samples of populations (Gray & Debreceny, 2014).

The level of sophistication of the tests applied by CAAT’s depends on the level of sophistication of the software used (Lanza et al., 2007). To make the step to more advanced tests additional tools should be used or new algorithms should be built into existing CAAT’s. For this research is chosen to use Rapidminer as data mining tool. Rapidminer is one of the most popular programs for data mining applications (Prekopcsak et al., 2011). It provides an integrated environment for machine learning and data mining. There are also various open-source plug-ins available which extend the possibilities of the program. For this research the ‘Anomaly detection extension’ is used as published in Goldstein & Uchida (2016). This plug-in contains various types of anomaly detection algorithms that can be applied in the audit domain. The details about the machine learning algorithms will be discussed in chapter 3.

2.6 Capabilities of the auditor

The level of sophistication of the tests that can be applied by CAATs also depends on the technical and IS related skills and capabilities the auditor (Lanza et al., 2007). Auditors have the domain knowledge about accounting rules and regulations, however, their IS related skills have not kept pace with the developments in accounting information systems (Curtis et al., 2009).

Due to this increased complexity and importance of information system knowledge there are also special IT auditors involved in every audit engagement. However, they often have a very different academic background and therefore limited accounting knowledge (Curtis et al., 2009). It would therefore be interesting to provide an easy to use model that general auditors can use in their analysis in addition to, for example, Excel. Rapidminer is, with its clean user interface, an easy to learn tool (Prekopcsak et al., 2011) and can be used by general auditors to improve the sophistication of their analysis and therefore the efficiency and effectiveness in the audit.

2.7 Conclusions

In this chapter, the business environment is investigated. Direct assessment of transactional data becomes more important in the financial audit of a company due to regulatory changes. Increase in the sophistication of the data analysis can improve efficiency and effectiveness of auditors. Investigating 100% of the data and flagging anomalous transactions will guide auditors in their control tasks. An easy to use technique would fit best with the capabilities of the general auditor. In the next chapter, background knowledge about anomaly detection will be gathered and appropriate algorithms for this research will be selected.
Chapter 3: Anomaly detection algorithms and the link to transaction level audit data

In this chapter appropriate theories and methods are identified out of existing literature. Anomaly detection is chosen as appropriate data mining technique as it is used to enhance traditional rule-based systems in other domains. An anomaly is both rare and deviates from the other examples in the dataset. There are four key components associated with anomaly detection that need to be addressed: the nature of the input data, the type of anomaly to be detected, label availability and the type of output that is expected. These aspects are all discussed in relation to transaction level data. Afterwards, three well-suited algorithms are selected from a taxonomy of anomaly detection algorithms. Finally, each selected algorithm is briefly described.

The chapter is structured as follows. Anomaly detection is introduced in paragraph 3.1. In paragraph 3.2, the key components of an anomaly detection problem are discussed and in paragraph 3.3, the anomaly detection algorithms are selected.

3.1 Introduction and definitions of anomaly detection

Anomaly detection algorithms are now often used to enhance traditional rule-based systems in other domains (Goldstein & Uchida, 2016). The terms outlier detection and anomaly detection are often used interchangeably (Chandola, Banerjee & Kumar, 2009) and the term anomaly is also commonly substituted by outlier, exception or deviation (Hadzic, Dillon & Tan, 2007). In this research the terms anomaly and anomaly detection will be used.

There has been a changing point in the focus and motivation for the detection of anomalies in data (Goldstein & Uchida, 2016). Before 2000 the main reason for anomaly detection was removing anomalous cases as part of data cleansing. After this point researchers started to get more interested in the anomalies itself since they are often associated with interesting events. The anomalies might be induced in the data for a variety of reasons but they have in common that they are “interesting to the analyst” (Chandola et al., 2009, p.2). This interestingness or real life relevance is a key feature of anomaly detection. In this research, the analyst is the auditor of a company.

Definition of an anomaly

Two main assumptions apply in anomaly detection (Goldstein & Uchida, 2016):

- Anomalies are different from the norm with respect to the value of their features.
- The anomalies are rare in a dataset compared to normal instances.

The term anomaly detection is not well defined from a mathematical point of view (Hofmann & Klinkenberg, 2013) because the characteristics of an anomaly depend on the specific problem type. Singh & Upadhyaya (2012) define anomalies as data instances that do not conform with a well-defined normal course of business. Based on the previous definitions an anomaly in this study is defined as:

“An anomaly is a data instance that is rare in the dataset compared to normal instances and does not conform with a well-defined normal course of business.”
The fact that the anomalies are rare and have deviating feature values compared to normal instances makes it possible for anomaly detection algorithms to identify them in a dataset.

The term ‘exception’ is in this report used to describe a deviating transaction that is interesting to an auditor to be detected (i.e. an anomaly with business value).

Visual example of anomaly detection

Figure 5 shows a 2-dimensional example of anomalies in a dataset. Region N1 and N2 are the normal regions in which most observations occur. Points that lay outside these regions can be classified as anomalies. These are the points A1, A2 and points in region A3. Figure 5 shows already that anomalies can occur in different ways. The different kinds of anomalies will be discussed in paragraph 3.2.
3.2 Key components associated with anomaly detection techniques

An anomaly detection problem has different aspects that need to be considered to define the appropriate anomaly detection technique. The important characteristics are discussed in this paragraph and linked to the problem of finding anomalies in transaction level audit data. Chandola et al. (2009) provide a framework the key components associated with the choice of an anomaly detection technique (Figure 6).

As shown in Figure 6, the anomaly detection technique connects the research area and the application domain. Anomaly detection is applied in various application domains, from medical applications to cyber intrusion detection (Chandola et al., 2009). The problem characteristics depend on the characteristics of the application domain. The different aspects that need to be discussed before selecting the appropriate anomaly detection technique are the nature of the data, availability of labeled data, type of anomalies that need to be detected and the type of output the technique provides. The different problem characteristics are discussed in the next paragraphs.

3.2.1 Nature of the input data

An important aspect of any anomaly detection technique is the nature of the input data. The input data consists of a collection of data instances. For example, a set of bank transactions made by a company. Each data instance consists of a set of features. These features describe the important characteristics of the data instance. Each bank transaction has for example a feature which shows the amount that is transferred, the name of the company to which it is transferred and the bank account number of this company.

Each feature can have his own type, such as:

- Binary
- Continuous
- Categorical
The type of the attributes determines the applicability of the anomaly detection technique. Not every technique supports both continuous and categorical variables. Another characteristic is that anomaly detection algorithms in general assume there is no relationship among the input cases (Tan et al., 2005). For example when anomalies occur over time this will not be easily detected by an anomaly detection technique. This will be further discussed in the next paragraph.

### 3.2.2 Type of anomaly

An important aspect to consider is the type of anomaly to be detected. Anomalies can be classified into three categories (Chandola et al., 2009).

- **Point anomalies**
- **Contextual anomalies**
- **Collective anomalies**

A point anomaly is the simplest type of anomalous record and also the focus of the majority of the research on anomaly detection. If an individual data instance can be considered as anomalous with respect to the rest of the data, this case is called a point anomaly. An example is given in Figure 5. The point $A_1$ is anomalous with respect to feature $Y$ since it has a much higher value compared to all other data instances. Chandola et al. (2009, p. 7) give an real-life example:

“Let the data set correspond to an individual’s credit card transactions. For the sake of simplicity, let us assume that the data is defined using only one feature: amount spent. A transaction for which the amount spent is very high compared to the normal range of expenditure for that person will be a point anomaly.”

The second category are contextual anomalies. A contextual anomaly is a data instance that is only outlying in a specific context, but not otherwise. Chandola et al. (2009, p. 7) again provide an example of a real-life case:

“A contextual attribute in the credit card domain can be the time of purchase. Suppose an individual usually has a weekly shopping bill of $100 except during the Christmas week, when it reaches $1000. A new purchase of $1000 in a week in July will be considered a contextual anomaly, since it does not conform to the normal behavior of the individual in the context of time even though the same amount spent during Christmas week will be considered normal.”

In comparison to the rich literature on point anomaly detection techniques, the research on contextual anomaly detection has been very limited. One often used technique is to transform the contextual anomaly problem to a point anomaly problem and apply regular point anomaly techniques (Chandola et al., 2009). The context of the anomaly could be included in the problem by using a ‘contextual attribute’ in addition to a ‘behavioral attribute’. When looking back again to Figure 5, the points in region $A_3$ are only anomalous when considering both feature $X$ and $Y$. In this way a contextual anomaly can also be detected with regular point anomaly detection techniques.

The third anomaly type is a collective anomaly. Collective anomalies are a set of data instances that are not anomalous themselves, but their occurrence together as a group is anomalous. A collective anomaly in the audit domain could be double payments of invoices. If the payment of an invoice only occurs once, this is not anomalous. But when the same invoice is paid multiple times this behavior as a group is anomalous. Collective anomalies can occur only in data sets in which data instances are related. As previously stated, almost all anomaly detection techniques assume there is no relation between data instances. Therefore the techniques used for collective anomalies are different and research on this type is very limited (Chandola et al. 2007).
3.2.3 Data labels
Anomaly detection in auditing can operate in three ways with respect to label availability (Thiprungsri & Vasarhelyi, 2011). A label is a binary classification of a data instance. In anomaly detection such a label represents if a case is normal or anomalous. Figure 7 shows the different methods: supervised, semi-supervised and unsupervised.

Supervised anomaly detection assumes the availability of labels in a training dataset. A predictive machine learning model is trained with this data. After the model is trained on labeled data, the model can be applied to unlabeled data and predict whether a new data instance is normal or anomalous. However, labels for transactional data are in practice not available. In addition, anomalous behavior is often dynamic in nature (Chandola et al. 2009). New types of anomalies might show up that have not happened before. It can in practice also happen that due to changes in the environment anomalous data points not anomalous later on. For example, the definition of a high amount paid might change when a business grows over time. Moreover, a supervised method is in practice not realistic for the analysis of transactional data (Jans et al, 2007).

The second type of machine learning is semi-supervised. In semi-supervised learning is assumed the training dataset that is used to train the machine learning model only consists of normal instances. However, it is not possible to verify the absence of error or anomalies in datasets with thousands of records (Bay et al. 2002).

A realistic type of machine learning for the analysis of transaction level audit data in practice is unsupervised machine learning. The transaction level data for financial audit typically has no labels (Jans et al., 2007). In addition, with unsupervised methods you can detect anomalies that have not occurred before which covers the dynamic nature of anomalous behavior. As you can see in Figure 7, there is no training set needed. In unsupervised machine learning the decision of the algorithm is based on intrinsic information of the dataset (Amer & Goldstein, 2013). An assumption made by the algorithm is that normal instances occur far more frequent than anomalies which is very realistic.

![Figure 7: Different anomaly detection modes depending on the availability of labels in the dataset](based on Goldstein & Uchida, 2016)
3.2.4 Output of anomaly detection algorithm
The way in which the anomaly detection algorithm classifies the output is the final aspect of an anomaly detection technique to be taken into account. Chandola et al. (2009) distinguish two types of output; a label or a score.

In the first category, the anomaly detection technique provides a label to each data instance. So each case is either normal or anomalous. The second approach is to score to each data instance. Scoring techniques assign a score based on the degree to which that case is considered an anomaly. In this way the output of such a technique is a ranked list of anomalies. The analyst can subsequently choose to either analyze the top X anomalies or use a cut-off threshold based on the anomaly score.

3.2.5 Conclusions
The anomaly detection algorithm that fits best to the problem type of transaction level audit data has the following characteristics:

- Unsupervised anomaly detection algorithm
- Capable of handling both categorical as continuous values
- Point anomaly detection algorithm
- Anomaly score output

There can be concluded that unsupervised techniques are the most realistic type of machine learning to apply on transaction level audit data. In practice, transactions do not have a label and can show up in different ways which may not all be present in a training set.

The features of interest in this study include both categorical as continuous variables. For example the amount of a transaction is continuous and the bank account number is a categorical variable. The anomaly detection technique should therefore support these input types.

The type of anomalies on which this study focuses are mainly point anomalies. This type of anomalies is the focus area of almost all research on anomaly detection and therefore multiple algorithms are available. It is still possible to utilize point anomaly detection algorithms for contextual anomaly problems. In order to do so, the context should be included in the data. In this way the contextual anomaly task can be transformed into a point anomaly detection problem (Goldstein & Uchida, 2016).

The scoring output is an ideal character of the anomaly detection techniques for our purposes. It allows the expert to use a domain and problem specific threshold to examine. When a technique uses a binary label the threshold needs to be pre-specified and does not allow the expert to directly make such a choice. For our research is therefore focused on techniques that provide an anomaly score.
3.3 Selection of the anomaly detection algorithms

The key components associated with an anomaly detection technique are discussed in the previous paragraph. In addition, for every component is discussed how it applies to the domain of our study. After gaining this knowledge there can be continued with the selection of specific anomaly detection algorithms that are most relevant for our research on transaction level audit data.

3.3.1 Taxonomy of unsupervised anomaly detection

Goldstein & Uchida (2016) provide a taxonomy of unsupervised anomaly detection algorithms (Figure 8). The algorithms can be classified into five main groups; nearest-neighbor based, clustering based, statistical, subspace based and classifier based. The algorithms represented in the taxonomy are based on practical application in the past and attention in the scientific community. All algorithms in this taxonomy are point anomaly detection algorithms and provide an anomaly score as output. The green boxes in Figure 8 show the algorithms selected for this research. The selection criteria will be discussed in the next paragraph. All machine learning algorithms are available as an open source plug-in of the Rapidminer data mining application.

![Unsupervised Anomaly Detection Algorithms](image)

Figure 8: Taxonomy of unsupervised anomaly detection algorithms (based on Goldstein & Uchida, 2016)

3.3.2 Selection of anomaly detection algorithms

In this paragraph the appropriate algorithms from the taxonomy in Figure 8 are chosen.

The first thing that should be decided based on the taxonomy in Figure 8 is whether the anomaly detection problem can be classified as either global or local. Goldstein & Uchida (2016) differentiate between global and local anomaly detection algorithms. This distinction is not included in the framework of Chandola et al. (2009) that is discussed in the previous paragraph. Looking back at Figure 5, A1 and A2 are both anomalies. However, A2 is considered a local anomaly because it lies very close to the normal cases in region N2. Point A1 is considered to be a global anomaly because it does not lay close to any other normal region. In our research is focused on global anomaly detection algorithms.

The histogram based outlier score (HBOS) is a simple statistical technique that assumes independence of the features. However, combined addressing multiple features is one of the reasons to use anomaly detection instead of simple statistics. The approach is fast but at the cost of precision (Goldstein & Uchida, 2016). Overall, HBOS is therefore not considered to be useful.

The subspace based algorithm (rPrincipal Component Analysis) in this program does not work with categorical variables and is therefore not used. The uCBLOF algorithm is the best performing cluster based algorithm (Goldstein & Uchida, 2016) and therefore the cluster based CBLOF and CMGOS are
not used. The remaining algorithms (k-NN, uCBLOF and One-class SVM) are considered to be useful and are further discussed one by one in the next paragraph.

3.3.3 Discussion of selected algorithms

The three selected algorithms are discussed in this paragraph to show how they work and why they are suitable for the application in this study. All selected techniques do not assume any distribution in the data as, for example, statistical outlier detection techniques do. This is an advantage above usual outlier detection methods. In addition, the selected algorithms are able to use both categorical as continuous features which is not possible using statistical outlier detection methods.

k-Nearest Neighbor algorithm (k-NN)

As the name implies, the k-Nearest Neighbor algorithm identifies for each point in the dataset the k points that are closest to itself. The algorithm assumes the normal data instances occur in dense neighborhoods, while anomalies occur far from their closest neighbors. A nearest neighbor based techniques does not make any assumption regarding the distribution of the data. The k-NN algorithm is a distance based technique. The mixed Euclidean distance is used as distance measure to be able to use both continuous and categorical variables. The distance between two point \( p = (p_1, p_2, \ldots, p_n) \) and \( q = (q_1, q_2, \ldots, q_n) \) in the ‘Euclidean n-space’ is calculated via the following formula (Greenacre, 2008):

\[
d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2}
\]

In which:

\( d(p, q) = \text{distance between point } p \text{ and } q \)

\( n = \text{number of dimensions} \)

The distance for the categorical variables is calculated via a dissimilarity coefficient (Greenacre, 2008).

The anomaly score of a point in the k-NN algorithm is the average of the distance to the nearest neighbors as proposed by Anguilli & Pizzuti (2002). The absolute value of the score depends very much on the dataset itself, the number of dimensions and on normalization (Goldstein & Uchida, 2016). However, when normalizing all features between 0 and 1, the anomaly score will also be between 0 and 1. A 1 indicating a very anomalous data point and a 0 indicating a data point in a dense region.

unweighted Cluster Based Local Outlier Factor (uCBLOF)

Clustering is the process of arranging similar data points into groups. Clustering based anomaly detection algorithms work with the output of a clustering algorithm. In this study is chosen to use the k-Medoids clustering algorithm as input for the anomaly detection algorithm to be able to use both categorical as continuous variables.

The uCBLOF algorithm assumes that anomalies lie in sparse and small clusters far from their cluster centroid. A heuristic is used by the uCBLOF algorithm to classify the clusters made by the k-medoids clustering into large and small clusters. Afterwards, an anomaly score is computed based on the distance of each data point to its cluster center based on the following formula (Amer & Goldstein, 2012):
Unweighted CBLOF \((p) = \begin{cases} \min \left( d(p,C_j) \right) & \text{if } p \in SC \text{ where } C_j \in LC \\ d(p,C_i) & \text{if } p \in C_i \in LC \end{cases}\)

With:
\(d = \text{(mixed euclidian) distance}\)
\(p = \text{data point}\)
\(C_j = \text{cluster center of cluster } j\)

\(SC = \text{small cluster}, LC = \text{large cluster}\)

The anomaly score computed by the uCBLOF varies between 0 and 1 if all features are normalized between 0 and 1. Cluster based anomaly detection algorithms have in theory lower computational demand which could make them attractive when analyzing large datasets (Amer, et al., 2013).

One-class Support Vector Machine (one-class SVM)
Support vector machines have been one of the most successful machine learning techniques for the past decade (Amer et al., 2013). SVM algorithms are often used as supervised or semi-supervised algorithms. An SVM algorithm separates the data points in classes using an optimal hyperplane between the normal and anomalous data points (Schölkopf, Platt, Shawe-Taylor & Smola, 2001). SVM based algorithms are unsupervised methods when using a soft margin (Amer et al., 2013) and show good results in anomaly detection compared to other algorithms (Goldstein & Uchida, 2016). The unsupervised one-class SVM anomaly detection is trained on the dataset and afterwards each data point is scored. The outlier score for the one-class SVM is calculated based on the following formula:

\[ f(x) = \frac{g_{max} - g(x)}{g_{max}} \]

With:
\(x = \text{data point}\)
\(f(x) = \text{anomaly score}\)
\(g(x) = \text{directed distance to decision boundary}\)
\(g_{max} = \text{maximum directed distance between dataset points and decision boundary}\)

By dividing between \(g_{max}\) the points on the decision boundary have a score of 1.0. Points with a score higher than 1 are therefore potential anomalies. The higher the score, the more anomalous the data point.
Chapter 4: Point anomaly detection model

In this chapter one application of unsupervised machine learning is proposed, tested and evaluated. The model is able to detect sales transactions with anomalous values. An anomalous value can be, for example, a very high profit of one transaction compared to the other transactions of that company. A second example is an entry user that only enters a transaction once. Auditors are interested in these transactions because these exceptional values can represent high risks or deviations from normal processes. The three anomaly detection algorithms (k-NN, uCBLOF and one-class SVM) selected in the previous chapter are applied and evaluated. By making use of these unsupervised machine learning algorithms, multiple features can be tested together in one analysis. This fast and easy to use test provides the domain expert, the auditor, with a list of transactions with an anomaly score. The most anomalous transactions are on top of the list and auditors can decide to analyze the top X transactions of the list, based on their expert opinion.

The chapter is structured as follows. First, the importance of the audit of sales transactions is discussed shortly. Afterwards, the dataset used for this analysis is described. For quantitative evaluation, the data is labeled and a model is built in Rapidminer. The performance of the different algorithms is compared and conclusions are drawn. Finally, five semi-structured interviews are performed with auditors to evaluate the model and identify how it can be used in practice.

4.1 Problem description & unit of analysis

The main stream of revenue for a production company arises from the sale of their products to their customers. Because of the fact that sales produce the main income for a company it is very important that every aspect concerned with the sales process is reported correctly. Johnstone, Gramling & Rittenberg (2013, p. 382) even state the following:

“Sales transactions are always material to a company’s financial statements and often are subject to manipulation. Many audit failures have been characterized by misstatement of sales. Because sales are often subject to misstatement, special attention is paid to the control environment and to management’s motivation to ‘stretch’ accounting principles to achieve desired reporting.”

In addition, the choice is based on three unstructured interviews within the public audit firm in which this study is conducted (Appendix B). Sales transactions have characteristics that fit with the potential of anomaly detection algorithms. There are big groups with the same type of transactions in which only a few deviations occur. Overall, sales transactions are chosen as unit of analysis because they are interesting to the auditor and anomaly detection algorithms could add value.

The anomaly detection model in this chapter focusses on the detection of transactions with anomalies in at least one feature (i.e. outliers). Anomaly detection algorithms have added value here compared to normal statistical outlier techniques in multiple ways. The algorithms are distance based. So, in the detection of anomalies the algorithms do not make any assumption about the distribution of the underlying data (which is usually unknown). In addition, the anomaly detection techniques are able to take multiple features into account in one test. Also the combined use of categorical and continuous variables is not possible with simple statistical outlier techniques.
For this research a real life dataset from an ERP system is used. The data is extracted from a SQL database. The dataset that is used consists of information regarding the sales process of a production company. The data consists of records (i.e. rows) and every record represents the sale of one product. Every record has multiple features (i.e. columns) that represent the characteristics of that transaction. The original dataset consists of 92 features and 7696 records. This dataset is selected because several interesting exceptions are found in an initial investigation of the data. The existing exceptions make it possible to evaluate the anomaly detection model performance.

4.2.1 Feature selection criteria

Selection criteria are set to determine the dataset that will be used for the analysis. The order type ‘standard orders’ are selected because this is the majority of the orders (90%) so they are most interesting for data analysis. Only the ‘standard order’ type is selected because otherwise the exceptions found can be due to deviations in the type of order. Only the year 2015 is selected because data of the full year is available. After the selection the data consists of 6501 records.

The features that are selected from the dataset are initially based on the identified risks as described in paragraph 2.2. The selected features can be linked to an accounting rule or current audit check (Table 3). An outlying or infrequent occurring value in the selected features should be interesting to an auditor to investigate further. The more outlying or less frequent the value, the more interesting it is. In five semi-structured interviews that are conducted, it is confirmed that the selected features are interesting and relevant (Appendix D). All respondents agreed that in their current analysis they also consider these features. No missing values are identified in the data.

The assessment of the dataset results in the selection of nine features. Based on these nine features, three new features are generated. This process is described in Appendix C. Finally, this results in eight features to be used in the anomaly detection model. The features are shown in Table 3. ‘Cost of goods sold’ represents the cost of making a product. ‘Revenue’ is the sales value of the product. ‘Profit’ is equal to the difference between revenue and cost of goods sold. The ‘Profit Percentage’ is the relative profit with respect to the costs. ‘Entry time’ is the time of the day the transaction is posted. ‘Document date-entry date’ represents the time that elapse between the original document date and the day the transaction is entered into the system. ‘Delivery surplus’ shows an amount if there is more delivered than ordered.

<table>
<thead>
<tr>
<th>#</th>
<th>Attribute</th>
<th>Attribute type</th>
<th>Reason to select feature</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cost of goods</td>
<td>Continuous</td>
<td>Large and infrequent transactions are interesting to investigate more throughout</td>
<td>- Boynton &amp; Johnson (2006)</td>
</tr>
<tr>
<td></td>
<td>sold</td>
<td></td>
<td></td>
<td>- Argyrou (2013)</td>
</tr>
<tr>
<td>2</td>
<td>Revenue</td>
<td>Continuous</td>
<td></td>
<td>- CAQ (2008)</td>
</tr>
<tr>
<td>3</td>
<td>Profit</td>
<td>Continuous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Profit %</td>
<td>Continuous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Entry time</td>
<td>Interval</td>
<td>Transactions posted on odd times of the day</td>
<td>- CAQ (2008)</td>
</tr>
<tr>
<td>6</td>
<td>Document date-</td>
<td>Integer</td>
<td>Long lapse of days between original document date and entry date</td>
<td>- CAQ (2008)</td>
</tr>
<tr>
<td></td>
<td>Entry date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Delivery surplus</td>
<td>Continuous</td>
<td>Transactions in which much more is delivered than ordered are interesting to investigate</td>
<td>- Markham (2015)</td>
</tr>
<tr>
<td>8</td>
<td>Entry user</td>
<td>Categorical</td>
<td>Entered by infrequent user</td>
<td>- CAQ (2008)</td>
</tr>
</tbody>
</table>

Table 3: Selected/Generated features for anomaly detection model
4.2.2 Defining exceptions to be found by the anomaly detection algorithms

An overview of the descriptive statistics of all 8 features is given in Appendix E. Every selected feature contains some outlying values. The distribution of the feature values is shown in Figure 9 to 16 on the next page. For example, in Figure 9, the feature ‘cost’ is visualized and there can be seen that out of the 6501 transactions, there are some transactions that deviate much from the rest of the data. Automatically detecting these values would be advantageous for auditors.

For each feature, a boundary is set that describes the range of normal values for that feature. Values outside these boundaries are both rare and deviate from the norm. A transaction with at least one value outside these boundaries is assumed to be an exception. An overview of the features with their normal and anomalous values is shown in Table 4. In total, 100 transactions have at least one value outside the boundaries, so the total number of exceptions is assumed to be 100. The total number of transactions is 6501. So, this is equal to \( \frac{100}{6501} \cdot 100 = 1.5\% \) of all transactions in the dataset. Argyrou (2012) tests in their analysis of ‘suspicious’ transactions a model with 1%, 3% and 5% of ‘suspicious’ transactions. Therefore the 1.5% in this analysis seems a realistic percentage.

The boundaries are set based on visual inspection of the data and discussed with auditors. There are no strict boundaries for a data point to be interesting for an auditor in real life. However, the fact that a data point is rare and deviates from the other data points is interesting. Therefore, the current boundaries are assumed to be reasonable for this analysis.

The dataset should be labeled to be able to quantitatively evaluate how the different algorithms perform. A label is a binary classification of a transaction. A transaction can be either normal or anomalous. A normal transaction is in this study represented by a 0 and an transaction with an anomalous value in at least one feature is represented by a 1. However, the initial dataset has no labels. In appendix G is described how the data is labeled. The labeling of the dataset enables the quantitative evaluation of the anomaly detection algorithms. A usual method to evaluate anomaly detection models is the true positive rate for the top k results in a data set that contains k outliers (Schubert, Wojdanowski & Zimek, 2012). There is assumed that the dataset contains 100 exceptions. Therefore the top 100 results will be examined and the true positive rate in this top 100 will be calculated.

A second and more advanced method of evaluation is the receiver operating characteristic (ROC) curve. This method plots the true positive rate against the false positive rate. This provides a nice visualization of the performance of the algorithms. To numerically compare the ROC plots, the area under this curve (AUC value) will be used (Schubert et al., 2012). All three evaluation types will be used to get a good overview of the model performance.
Table 4: Definition of normal and exceptional values for sales transaction features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Normal / frequently occurring value(s)</th>
<th>Number of transactions outside boundary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>Value: 0 - 500.000</td>
<td>8 exceptional transactions</td>
</tr>
<tr>
<td>Revenue</td>
<td>Value: 0 - 1,000,000</td>
<td>3 exceptional transactions</td>
</tr>
<tr>
<td>Profit</td>
<td>Value: -400.000 - 400.000</td>
<td>8 exceptional transactions</td>
</tr>
<tr>
<td>Profit %</td>
<td>Value: -100 - 100</td>
<td>7 exceptional transactions</td>
</tr>
<tr>
<td>Delivery surplus</td>
<td>Value = 0</td>
<td>15 exceptional transactions</td>
</tr>
<tr>
<td>Entry user</td>
<td>User with #transactions &gt; 30</td>
<td>57 transactions by 4 infrequent users</td>
</tr>
<tr>
<td>Entry time</td>
<td>7:00 o’clock – 20:00 o’clock</td>
<td>12 exceptional transactions</td>
</tr>
<tr>
<td>Difference document date – entry date</td>
<td>Value = 0</td>
<td>2 exceptional transactions</td>
</tr>
</tbody>
</table>

Total 100 individual exceptional transactions
4.3 Modeling in Rapidminer

The selected anomaly detection algorithms are available in an open source extension for the Rapidminer data analytics software. In Rapidminer the different building blocks are connected in order to generate the right input for every process step. In this way a model is created that loads an Excel file into Rapidminer and automatically (by making use of the anomaly detection algorithm) generates a list with the most anomalous transaction at the top, followed by a ranked list based on decreasing anomaly score.

4.4 Test design

With a labeled dataset and the evaluation methods described in paragraph 4.2, the three algorithms (k-NN, uCBLOF and one-class SVM) can be applied and tested. For unsupervised algorithms there is no exact measure to determine the parameter setting. Therefore is for every algorithm checked how it performs for different parameter settings. The parameter settings are chosen as shown in Table 5. For each algorithm 5 different parameter values are tested. In this way, it can be determined how sensitive the algorithms are to the different parameter settings.

Table 5: Test design

<table>
<thead>
<tr>
<th>Parameter values</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN [10,20,30,40,50]</td>
<td>Based on the rule of thumb mentioned in Goldstein &amp; Uchida (2016)</td>
</tr>
<tr>
<td>uCBLOF [10, 20, 30, 40, 50]</td>
<td>Based on the ‘elbow rule’ (Leskovec &amp; Rajaraman (n.d.))</td>
</tr>
<tr>
<td>One-class SVM [0.05, 0.1, 0.2, 0.3, 0.5]</td>
<td>Based on the tests in Schölkopf et al. (2001)</td>
</tr>
</tbody>
</table>

4.5 Results

The results of the different algorithms are shown in Table 6. For every algorithm the true positive rate in the top 100 results is given. In addition, the AUC value for every model is shown. For the optimal parameter values per algorithm, the ROC-curve is plot in Figure 17. The results will be discussed in paragraph 4.6.

It should be noted that the anomaly detection algorithms and the k-medoids clustering algorithm are non-deterministic (Goldstein & Uchida, 2016). This means the algorithms could generate different results every run. However, as shown in the research of Goldstein & Uchida (2016) the algorithms perform very deterministic. During our tests, different runs always provided the same outcome. Therefore is assumed the algorithms provide deterministic output and showing the output of one run is sufficient.

Table 6: Performance of different anomaly detection algorithms

<table>
<thead>
<tr>
<th>k-NN</th>
<th>TPR-top 100 results</th>
<th>AUC value</th>
<th>uCBLOF</th>
<th>TPR-top 100 results</th>
<th>AUC value</th>
<th>One-class SVM</th>
<th>TPR-top 100 results</th>
<th>AUC value</th>
</tr>
</thead>
<tbody>
<tr>
<td>k=10</td>
<td>91%</td>
<td>0.9912</td>
<td>k=10</td>
<td>70%</td>
<td>0.9417</td>
<td>nu=0.05</td>
<td>85%</td>
<td>0.9589</td>
</tr>
<tr>
<td>k=10</td>
<td>94%</td>
<td>0.9900</td>
<td>k=20</td>
<td>72%</td>
<td>0.9631</td>
<td>nu=0.1</td>
<td>85%</td>
<td>0.9667</td>
</tr>
<tr>
<td>k=10</td>
<td>93%</td>
<td>0.9900</td>
<td>k=30</td>
<td>76%</td>
<td>0.9676</td>
<td>nu=0.2</td>
<td>86%</td>
<td>0.9709</td>
</tr>
<tr>
<td>k=10</td>
<td>91%</td>
<td>0.9900</td>
<td>k=40</td>
<td>83%</td>
<td>0.9724</td>
<td>nu=0.3</td>
<td>85%</td>
<td>0.9682</td>
</tr>
<tr>
<td>k=10</td>
<td>91%</td>
<td>0.9891</td>
<td>k=50</td>
<td>72%</td>
<td>0.9632</td>
<td>nu=0.5</td>
<td>83%</td>
<td>0.9646</td>
</tr>
</tbody>
</table>
4.6 Evaluation

The proposed model is evaluated in two ways. First, a data based review is done based on the output of the model. In addition, the model is evaluated with five auditors to investigate how the model can make their work more effective and efficient.

Data based

Overall, all algorithms detect most of the labeled exceptions. The best results for k-NN, one-class SVM and uCBLOF are respectively 94%, 86% and 83% true positive in the top 100 results (with 100 exceptions in the dataset). There can be concluded that the k-Nearest Neighbor algorithm performs best in finding the defined exceptions in this test. There is analyzed which transactions are found instead of the labeled exceptions. These are transactions with feature values that are close to, but inside the boundary that is set for the defined exceptions. In addition, the uCBLOF and the one-class SVM find some ‘local’ outliers. For example in the feature ‘entry time’ in which are gaps between the end and beginning of an hour. This could of course be transformed but it indicates the sensitivity of the algorithms. In this case has the k-NN algorithm more intuitive results.

The parameter setting does not influence the results very much for the k-NN and the one-class SVM algorithm. This is a nice result for the use in practice, since the exact parameter setting is not easy determinable in advance. However, for the uCBLOF algorithm there is a bigger difference in the results of the different parameter values. This sensitivity to an optimal parameter value makes the algorithm less attractive for practical use.

There is investigated how the algorithms detect the exceptions in the different (type of) features. The true positive rate per features is shown in Appendix H. Interesting to notice is that all algorithms are able to detect the categorical variable very well. All algorithms score 100% true positive on the detection of the categorical variable ‘Entry User’. This has also interesting practical implications because many variables in the audit environment are categorical. Examples are company name, G/L account number, bank account number, etc.. When an typing error is made in for example a bank account number this will result in a bank account number that is only used once. The anomaly detection algorithm will be able to detect this infrequent occurring bank account number. It should be noted that for all transactions to companies to which also only once a year a transactions occurs
will have the same characteristics as the transaction with the typo. In practice, a lot of false positives will be the result.

It is also interesting to notice that the feature ‘delivery surplus’ has very low true positive rates. This can be explained by the distribution of the values in this feature. The feature ‘delivery surplus’ has some very big exceptions compared to the other exceptions in this feature. Therefore, when the feature values are normalized between 0 and 1, only the very big exception are found and the other exceptions not. So the results of the model will be influenced if there is a big range in the exceptional values in a feature. This is a drawback of the anomaly detection algorithms.

Evaluation with auditors
To evaluate whether the model is interesting for the end user, the auditor, five semi-structured interviews are conducted. The interviewees are people with different levels of experience at the financial audit department of the company at which this study is conducted. In this way both the open minded view of a trainee as the more experienced view of an audit manager are combined. The interviews took approximately 30 minutes in which 10 minutes were used to show the model and the model output. The remaining 20 minutes were used to discuss different aspects of the model along on a short questionnaire (Appendix D).

The auditors acknowledge that the model would be helpful to provide fast insights in the organization and their processes in an efficient way. It helps to map where the deviations from normal processes could be and it helps to identify risk areas. The model could be used as an additional tool beside the normal use of for example Excel. Also the program itself, Rapidminer, is rewarded because of its simplicity and nice visualizations. An example is given in Figure 18 in which is shown that 4 different users post transactions infrequently and one transaction has an extreme high value compared to the other transactions. The color represents the anomaly score.

![Figure 18: Visualization of Rapidminer output (cost vs. entry user)](image)

For the auditor, the most exceptional cases are most interesting. Therefore, the top 10 results of the k-NN, uCBLOF and SVM algorithm (with the optimal parameter value) are evaluated and shown in Appendix I. The top 10 of both the k-NN and the SVM algorithm include anomalous values in all 8 variables at least once. This shows these algorithms are balancing the different features. It is also interesting to notice that the highest ranked exceptions contain anomalies in multiple features. The auditors state during the interviews that it is most interesting to find transactions that have an anomalous value in multiple features since this would usually not show up in current tests.
To validate the model more specifically, the top 30 transactions of the k-NN algorithm are one by one assessed by two auditors. The transactions in the output of the model are shown and the auditors could assign the transaction to one of 4 categories. The categories are based on the reactions in the semi-supervised interviews (Appendix D). A transaction could be interesting by itself, for example due to extreme high value of a transaction and additional analysis of source documents may be required. Secondly, a transaction could be interesting because it indicates a possible deviation from a process that needs to be investigated. For example when the entry user of a transaction occurs very infrequent, an auditor is interested in the role of this person. Thirdly, a detected deviating transaction could be interesting because it provides insights in the organization but the auditor believes no action is needed. An example is a transaction with a relatively very high profit that is not in conflict with accounting rules but provides insights in the business of the client. Finally, the model could provide uninteresting transactions in the output. An overview is given in Table 7.

The results are shown in Appendix P. The average results in the assignment of transactions to the different categories is shown in Table 7. Overall, most transactions are assigned to the second category: check the underlying process to get more insight. The auditors are interested in how these deviations occur in the business. For example why there is a big time difference between posting of a transaction and the original document date, how it can happen that only very low costs are assigned to an order with a high revenue or what the role is of an user that posts very infrequent.

Table 7: Overview percentages of transactions assigned to different categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage of transactions assigned to category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interesting transaction: Check transaction details/ documents (with client)</td>
<td>8%</td>
</tr>
<tr>
<td>Interesting transaction: Check underlying process</td>
<td>83%</td>
</tr>
<tr>
<td>Interesting transaction: Provides insights but probably no action required</td>
<td>8%</td>
</tr>
<tr>
<td>Non interesting transaction</td>
<td>0%</td>
</tr>
</tbody>
</table>

It should be noted that the interest in specific transactions is related to the specific engagement. In different situation, different features and transactions can be of interest. This is also an aspect that follows from the interviews. As shown in Figure 3 (Chapter 2) there are four sources for the selection of processes and features to consider:

- Firms policies & procedures
- Audit templates
- Audit team brainstorming
- Individual auditor decisions

Therefore can be concluded that based on the situation, the selected and generated features that are interesting can vary. This is also an interesting aspect since the model is not limited to one specific problem area, but more broadly applicable. This applicability is discussed in paragraph 4.8.

4.7 Conclusion

There can be concluded that anomaly detection algorithms provide an interesting technique for the detection of outlying or infrequent occurring values, when considering multiple features of transactions together in one test. It should be noted that the anomaly detection algorithms detects only outliers (i.e. anomalies in one feature) in this test. However, when a feature has outlying values in multiple features, these transactions are highlighted by the model (Table 8). The top 5 transactions...
of the output of the uCBLOF algorithm is shown in Table 8. A ‘1’ is assigned to a feature if a transaction has an outlying value in that feature. As you can see, the top 5 results have an outlying value in multiple features. For example, the 3rd transaction in Table 8 has very high costs, very high value, is posted on a on a strange time (very late or early on a day) and by an infrequent entry user. It is interesting that especially these transactions with outlying values in multiple features are on top of the list for an auditor to review.

Table 8: Top 5 output uCBLOF anomaly detection algorithm

| uCBLOF (k=40) |  |  |  |  |  |  |  |  |
|----------------|---|---|---|---|---|---|---|
| Outlier top 5  | Cost | Value | Profit | Profit % | Delivery surplus | Entry user | Entry time | Document date-
| 1               | 0   | 1   | 1     | 0       | 0              | 1          | 0          | 0           |
| 2               | 0   | 1   | 1     | 0       | 0              | 1          | 0          | 0           |
| 3               | 1   | 1   | 0     | 0       | 0              | 1          | 1          | 0           |
| 4               | 0   | 0   | 0     | 0       | 0              | 1          | 1          | 0           |
| 5               | 0   | 0   | 0     | 0       | 0              | 1          | 1          | 0           |

4.8 Deployment of the model

To know when to apply the model, different moments of the audit process are identified (during the interviews) in which the model will be useful. Three main stages of the audit of a company are identified and mentioned by multiple auditors:

- Planning of audit: To get knowledge about the organization.
- Interim control: To identify deviations from processes.
- Year-end control: To identify ‘suspicious’ data instances.

In addition, there is investigated which other types of datasets could be analyzed by making use of unsupervised anomaly detection techniques. In the current model are sales transactions analyzed. During the interviews multiple other processes are identified that can benefit from unsupervised anomaly detection algorithms. These are:

- Audit of the purchasing process
- (Manual) Journal entry testing
- Bank payment organization (accounts receivable, accounts payable)

Important steps towards a more data driven audit can be made by applying the model in multiple stages of the audit process and to multiple processes.

![Figure 19: Overview anomaly detection process](image-url)
An process model of how the process will work in practice is shown in Figure 19. The process starts with a trigger from an auditor to analyze a certain type of transactions (sales transactions in the model in this chapter) to check for exceptions. The dataset is loaded into Rapidminer software (Rapidminer automatically checks for missing values). The relevant features of the dataset are selected and if needed, new features can be generated. The selected features are normalized such that all features have the same range. The parameter for the anomaly detection algorithm needs to be set (see Appendix F for explanation). The unsupervised machine learning algorithm is applied to the data and automatically generates a list with transactions and their features starting with the most anomalous transaction. The domain expert, the auditor, can analyze the top X transactions. If there are exceptional transactions interesting to the auditor, additional information can be investigated to determine if it is needed to follow up on the transaction. If the transaction is still exceptional it will be discussed with the client that is audited.

4.9 Extension of the model
In the current model, 8 different features are combined into one test. All feature values are normalized between 0 and 1 such that all variables are treated in the same way by the algorithm. It can however be the case that an auditor is especially interested in one specific feature. In other words, the auditor want to give a higher weight to one of the variables.

This extension is modeled in Rapidminer. After the data is normalized, a weight factor for each feature is made. By default the weights are set to 1.0. The auditor can easily adapt the weight of a feature.

An example is given for the feature ‘delivery surplus’. In Table 9, the top 10 results of the k-NN algorithm are given for different weights of the feature ‘delivery surplus’. An 1 represents a transaction with an anomaly in the ‘delivery surplus’ feature and an 0 means another anomaly is considered more important by the algorithm. There can be concluded that the anomalies in the feature of interest are more and more getting on top of the list when the weight is increased.

Table 9: Results for different weight factors of the feature delivery surplus

<table>
<thead>
<tr>
<th>K-NN ALGORITHM (K=20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEIGHT FACTOR</td>
</tr>
<tr>
<td>TOP 10 RESULTS</td>
</tr>
<tr>
<td>#1</td>
</tr>
<tr>
<td>#2</td>
</tr>
<tr>
<td>#3</td>
</tr>
<tr>
<td>#4</td>
</tr>
<tr>
<td>#5</td>
</tr>
<tr>
<td>#6</td>
</tr>
<tr>
<td>#7</td>
</tr>
<tr>
<td>#8</td>
</tr>
<tr>
<td>#9</td>
</tr>
<tr>
<td>#10</td>
</tr>
</tbody>
</table>
Chapter 5: Contextual anomaly detection

The model in the previous chapter focused on the detection of transactions with exceptional values in one dimension; an exceptionally high ‘profit margin’ or infrequent ‘entry user’. In this chapter a bit more complex situation is investigated. The unsupervised anomaly detection algorithms are able to detect points that are only anomalous in a specific context, i.e. contextual anomalies. This can be interesting in various tests within audit and advisory to find errors in transaction level data in a flexible and adaptive way without hard coding specific rules. This method is applied to four different situations. First, anomaly detection is applied to a two- and a three-dimensional problem based on synthetic data. In this way the idea of anomaly detection in audit is shown and the best performing anomaly detection algorithm is selected. Afterwards two anomaly detection problems based on real world data are identified, tested and evaluated. Overall, the tests get more complex step by step during this chapter. In this way the possibilities and limitations of anomaly detection in audit are identified.

This chapter is structured as follows. In the first paragraph an introduction is given to the application of contextual anomaly detection. In paragraph 5.2, three different anomaly detection algorithms are tested on two problems based on synthetic data. In 5.3, two tests are performed and evaluated based on real world data. In 5.4 is investigated if an learning effect could be included in the unsupervised models.

5.1 Introduction to contextual anomaly detection in audit

The focus of the proposed model in the previous chapter was on detecting exceptional values in various features. A transaction that the model detected was for example one with an exceptional ‘profit margin’ or an exceptional ‘entry user’. These exceptions occur in one dimension (e.g. in ‘profit margin’ and/or in ‘entry user’). It is however even more interesting to find anomalies that are exceptions only when considering multiple dimensions, i.e. contextual anomalies. An auditor would not find these transactions easily by making use of sorting in Excel or outlier detection in one dimension. This makes it an interesting line of research (Argyrou, 2012).

In this chapter is focused on the detection of contextual anomalies with at least one categorical variable. In Figure 20 is the theoretical example from chapter 3 shown. In practice, the data is however more noisy as for example shown in Figure 21 where the value of orders is set out against the profit. In this case only the extreme cases will be found by the anomaly detection model which are already analyzed in the model in chapter 4. Therefore is for the contextual anomalies in this chapter focused on situations with at least 1 categorical variable.

![Figure 20: Theoretical example two dimensional anomalies](image1)

![Figure 21: Practical example two dimensional data](image2)
An example of this situation is shown in Figure 22. In this figure a theoretical example is shown of a firm with five suppliers: Company A, Company B, Company C, Company D and company E. Each supplier has their own payment period in which they should pay their invoices. In Figure 22, the five companies are shown on the x-axis and the ‘days payment outstanding’ is plot on the y-axis. Each point represents an outstanding payment. What you see is that company A and D usually pay within 30 days. Clearly there are three strange points in the figure. For example, point 3 represents an invoice that is outstanding for 85 days where that company usually pays within 30 days.

![Figure 22: Synthetic example of contextual anomaly detection](image)

However, when you would consider only one dimension, in this case days payment outstanding, you would not find these points as outliers. For example, point 1 of company A would fall in the group of payment periods of company B and C if you only consider one dimension. It is necessary to consider two dimensions to find the three outlying points. This is an example of the problem type for which anomaly detection algorithms work very well. These unsupervised machine learning algorithms should be able to automatically detect the deviating points in the data. This application of anomaly detection is new to the audit field and could be an interesting first step towards a more automatic and flexible data-driven audit.

The same three anomaly detection algorithms as in chapter 4 are selected and compared in this chapter:
- K-Nearest Neighbor algorithm
- uCBLOF
- One-class SVM

For further development of unsupervised anomaly detection in the audit field, it is interesting to know which algorithm performs best and can thus can be applied in practice. For each algorithm the default parameter setting used since the algorithms are not very sensitive to the parameter settings.

### 5.2 Anomaly detection model with synthetic data

General anomaly detection research (e.g. Chandola et. al, 2007) indicates that the detection of contextual anomalies is generally not easy to solve. For example because the boundary between normal and anomalous behavior is not precise due to noisy data. Therefore, in this paragraph is investigated how anomaly detection performs on two problems based on synthetic data. A two dimensional problem is analyzed in the next paragraph. Afterwards, a three dimensional problem is investigated in 5.2.2.
5.2.1 Two dimensional anomaly detection problem
The problem described at the start of this chapter is used as the two dimensional anomaly detection
problem (Figure 22). The goal of the anomaly detection algorithm in this application is to detect
invoices that are not paid within the usual period for that specific company. The problem is related to
sales transactions and is identified during initial interviews with IT auditors (Appendix A).

Data description
Days payment outstanding is the number of days a customer takes to pay their invoice. Three common
used payment terms are payment between respectively 30, 60 or 90 days (Invoicing and payment
terms, n.d.). An overview of the data that is generated is shown in Table 10. For each company, 20
points are generated within a range related to their payment period. In total 100 points are simulated.
Three values for ‘days payment outstanding’ are changed to simulate an anomaly.

Table 10: Synthetic data, days payment outstanding per company

<table>
<thead>
<tr>
<th>Feature 1</th>
<th>Feature 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A</td>
<td>19x Random (uniform dis.) [25;30] + 1x 60 (anomaly)</td>
</tr>
<tr>
<td>Company B</td>
<td>20x Random (uniform dis.) [55;60]</td>
</tr>
<tr>
<td>Company C</td>
<td>20x Random (uniform dis.) [55;60]</td>
</tr>
<tr>
<td>Company D</td>
<td>18x Random (uniform dis.) [25;30] + 1x 57 + 1x 87 (anomalies)</td>
</tr>
<tr>
<td>Company E</td>
<td>20x Random (uniform dis.) [85;90]</td>
</tr>
</tbody>
</table>

Results
The data is normalized (between 0 and 1) and afterwards the three anomaly detection algorithms (k-
NN, uCBLOF and one-class SVM) are applied to the data. A summary of the results is shown in Table
11. The red color represents the exceptional data points and the green color the first normal data
point. The plot of the one-class SVM is shown in Figure 23. The other tables and visualizations of the
results are given in Appendix J.

Table 11: Anomaly score of different algorithms

<table>
<thead>
<tr>
<th>Top 4 outliers</th>
<th>Outlier score k-NN</th>
<th>Outlier score One-class SVM</th>
<th>Outlier score uCBLOF</th>
<th>Day payment outstanding</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.85</td>
<td>38.13</td>
<td>0.89</td>
<td>87</td>
<td>Comp. D</td>
</tr>
<tr>
<td>#2</td>
<td>0.48</td>
<td>38.08</td>
<td>0.54</td>
<td>60</td>
<td>Comp. A</td>
</tr>
<tr>
<td>#3</td>
<td>0.43</td>
<td>37.89</td>
<td>0.43</td>
<td>57</td>
<td>Comp. D</td>
</tr>
<tr>
<td>#4</td>
<td>0.03</td>
<td>2.68</td>
<td>0.04</td>
<td>25</td>
<td>Comp. D</td>
</tr>
</tbody>
</table>

Discussion
The results in Table 11 and Figure 23 show that all three algorithms are very well able to distinguish
between the normal data points and the anomalies. There is a big gap between the outlier score for
the first three points and the fourth, the normal transaction. Interesting to notice is the one-class SVM
algorithm provides almost the same outlier score to all three anomalies where the k-NN and the
uCBLOF give a higher score to the data point [company D, 87]. The explanation can be found in the underlying technique of the algorithms. The k-NN and the uCBLOF calculate the distance to the closest points or cluster center. The horizontal distance in Figure 23 (inter category distance) is the same for each anomaly since each anomaly lies close to one other group of data points. However, the vertical distance in Figure 23 (intra category distance) from the anomalies to the clusters of points differ. So, different than may be expected from Figure 23, the k-NN and the uCBLOF algorithm calculate the anomaly score based on the cluster of data points within the same category (vertical). This results in the different outlier scores of the anomalies. The SVM algorithm creates a higher dimensional space in which the distance of the anomalies to the decision boundary is the same and therefore the same anomaly score is assigned.

The interpretation of the value of the outlier score is important to evaluate. When adopting an anomaly detection algorithm should be clear which value indicates an anomaly and which indicates a normal data point. This is different for each of the three algorithms (as described in Chapter 4). For the one-class SVM algorithm, values around 1 are on the decision boundary and the higher the outlier score the more anomalous the data point. For k-NN and uCBLOF normal data points will get a value close to 0 and anomalous data points a value close to 1 (due to normalization between 0 and 1). For all algorithms, the outlier score relative to the rest of the data points provides the best indication on whether or not a point is an exception.

Related to this test, an anomaly detection algorithm could for example be implemented in an audit control tool to provide an alert when a supplier does not pay in time. It can also be used to indicate the risk related to non-paid invoices. This leads to a more flexible approach for internal audit.

### 5.2.2 Three dimensional anomaly detection problem

The previous approach can be extended towards a problem in three dimensions. In the problem that is addressed in this paragraph there is every week (feature: Date) a transaction of a certain amount (feature: Amount) to a certain supplier (feature: Supplier). Only once a payment is switched between two suppliers which could be detected by the anomaly detection algorithm.

#### Data description

In this problem there are two types of weekly payments (1.000 to supplier A and 5.000 to supplier B) for half a year. Total number of data points is 52. There are two anomalous transactions, namely, at 22-04-2016 the amount paid to both suppliers is switched. An overview of the synthetic data is given in Table 12.

**Table 12: Overview synthetic data with features ‘date’, ‘supplier’ and ‘amount’**

<table>
<thead>
<tr>
<th>Date</th>
<th>01-01-'16</th>
<th>08-01-'16</th>
<th>...</th>
<th>15-04-'16</th>
<th>22-04-'16</th>
<th>29-04-'16</th>
<th>...</th>
<th>24-06-'16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier</td>
<td>Supplier A</td>
<td>Supplier A</td>
<td>...</td>
<td>Supplier A</td>
<td>Supplier A</td>
<td>Supplier A</td>
<td>...</td>
<td>Supplier A</td>
</tr>
<tr>
<td>Amount</td>
<td>1.000</td>
<td>1.000</td>
<td>...</td>
<td>1.000</td>
<td>5.000</td>
<td>1.000</td>
<td>...</td>
<td>1.000</td>
</tr>
<tr>
<td>Supplier</td>
<td>Supplier B</td>
<td>Supplier B</td>
<td>...</td>
<td>Supplier B</td>
<td>Supplier B</td>
<td>Supplier B</td>
<td>...</td>
<td>Supplier B</td>
</tr>
<tr>
<td>Amount</td>
<td>5.000</td>
<td>5.000</td>
<td>...</td>
<td>5.000</td>
<td>1.000</td>
<td>5.000</td>
<td>...</td>
<td>5.000</td>
</tr>
</tbody>
</table>

The feature ‘amount’ is normalized between 0 and 1. It is important to notice that the date cannot be used in its original form. A data transformation step is required to use the feature ‘date’ in the anomaly detection algorithms. As shown in Appendix K, the algorithms do not provide any useful result when the original value for the date is used that is not normalized. The ‘date’ feature is transformed to an integer value and afterwards these values are normalized between 0 and 1.
Results

The three different anomaly detection algorithms are applied to the resulting dataset and the results are shown in Table 13. The red color in the table represents the anomaly score of the exceptions and the green color the anomaly score of the normal cases. In Figure 24, the results of the k-NN algorithm are plot with the date on the x-axis, amount on the y-axis and the color showing the anomaly score. The plots of the other algorithms are shown in Appendix L. The two red points have the highest anomaly score and this are the two exceptions in the dataset. The other points are blue and this refers to a normal data point.

Table 13: Output different anomaly detection algorithms

<table>
<thead>
<tr>
<th>Top 5 outliers</th>
<th>Outlier score k-NN</th>
<th>Outlier score One-class SVM</th>
<th>Outlier score uCBLOF</th>
<th>Algorithm</th>
<th>Date</th>
<th>Amount</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>1.0</td>
<td>35</td>
<td>1.0</td>
<td>k-NN</td>
<td>22-04-'16</td>
<td>5.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SVM</td>
<td>22-04-'16</td>
<td>5.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>uCBLOF</td>
<td>22-04-'16</td>
<td>5.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td>#2</td>
<td>1.0</td>
<td>35</td>
<td>1.0</td>
<td>k-NN</td>
<td>22-04-'16</td>
<td>1.000</td>
<td>Supplier B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SVM</td>
<td>22-04-'16</td>
<td>1.000</td>
<td>Supplier B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>uCBLOF</td>
<td>22-04-'16</td>
<td>1.000</td>
<td>Supplier B</td>
</tr>
<tr>
<td>#3</td>
<td>0.23</td>
<td>7.7</td>
<td>0.88</td>
<td>k-NN</td>
<td>24-06-'16</td>
<td>1.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SVM</td>
<td>24-06-'16</td>
<td>1.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>uCBLOF</td>
<td>01-01-'16</td>
<td>1.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td>#4</td>
<td>0.23</td>
<td>7.7</td>
<td>0.84</td>
<td>k-NN</td>
<td>24-06-'16</td>
<td>5.000</td>
<td>Supplier B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SVM</td>
<td>01-01-'16</td>
<td>1.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>uCBLOF</td>
<td>08-01-'16</td>
<td>1.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td>#5</td>
<td>0.22</td>
<td>7.7</td>
<td>0.80</td>
<td>k-NN</td>
<td>01-01-'16</td>
<td>1.000</td>
<td>Supplier A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SVM</td>
<td>24-06-'16</td>
<td>5.000</td>
<td>Supplier B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>uCBLOF</td>
<td>15-01-'16</td>
<td>1.000</td>
<td>Supplier A</td>
</tr>
</tbody>
</table>

Figure 24: Visualization results k-NN algorithm
Discussion

All three algorithms are able to detect the two anomalous data points in the dataset. The two anomalous records have the highest anomaly score in the output of each algorithm. However, the distinction between normal and anomalous records is less clear for the one-class SVM and the uCBLOF algorithm (see Table 13). The one-class SVM assigns an anomaly score of 7.7 to the two payments on the first date and the two payments on the last date (Remember that a score above 1 is usually an anomaly in the one-class SVM output). In this data view they are sort of anomalies too since they are the points at the boundary of the data. However, they are not interesting and this will require some additional programming when applying the algorithm in practice. The uCBLOF algorithm performs even less when looking to the anomaly score. The anomaly score gradually decreases and there is no clear distinction between the anomalies and normal instances. The k-NN algorithm outperforms in this case both the one-class SVM and the uCBLOF algorithm. The k-NN algorithm provides a clear distinction between the two anomalies and all other data points.

The reason for the difference in the performance can be found in the underlying technique of the algorithms. The clusters that result from the k-medoids clustering algorithm do not make much sense and therefore the outlier score of the uCBLOF is less relevant. The SVM transforms the data to a higher dimensional space which results in a proper detection of the anomalies. However, in this higher dimensional space, other points are also considered to be anomalous. The k-NN algorithm simply finds the closest data points in the neighborhood. Since each normal transaction has a lot of other transactions with the same amount, the distance between these data points is very small. Only the transactions with a deviating amount have a higher distance to their neighbors and therefore are detected as anomalies.

The detection of deviations from a standard pattern is very useful in practice in the financial control of a company. Many examples can be thought of in which it can be useful to control transactions. For example when a bank account number and a company number are always in used in the same combination, deviations could be detected automatically and prevent errors in (almost) real time.

It should be noted that the anomaly detection algorithm does not detect the anomalous behavior over time. If there is aimed to learn the model what a normal transaction for a company looks like, a form of supervised learning is advised. Anomaly detection algorithms just detect deviating data points because they are rare in the dataset.

5.2.3 Conclusion

Based on the previous two problems with synthetic data can be concluded that anomaly detection algorithms are promising for a more flexible and dynamic way of finding exceptions in audit data. In both theoretical problems they are able to detect the anomalous data instances without any prior coding. Based on the previous analysis the k-NN algorithm shows the best results due to clear difference in the anomaly score between the normal and exceptional data points. This makes the exceptions well distinguishable from the normal data points.
5.3 Anomaly detection model with real world data

The application of the anomaly detection algorithms on synthetic data showed good results. In this paragraph, two problems will be investigated with real life data. There will again be started with a two dimensional problem. Afterwards this two dimensional problem will be extended to a more complex analysis on a second real life dataset. For each real life application, a process model will be provided to show how to use the method in practice. In this chapter only the k-NN algorithm is used because previous paragraph showed it performed best.

5.3.1 Detection of debit (credit) postings to typical credit (debit) accounts

During the interviews conducted to evaluate the model in chapter 6 (Appendix D), multiple other application areas are identified where anomaly detection algorithms potentially could add value. One of those tests is to identify credit postings to an account on which is normally only debited. And the other way around: Identify debit postings to an account on which normally is only credited. In the accountancy literature this is also mentioned as a test in the manual journal entry process:

“Identify debits to typical credit accounts and credits to typical debit accounts” (CAQ, 2008).

This problem has the characteristics of a two dimensional anomaly detection problem. There are two features of interest: ‘Account number’ and ‘Debit/Credit indicator’. When a credit posting is made this is of course not special by itself. However, when a credit posting is made while that account is normally only debited it should get the attention of the auditor because this could indicate an error.

Data description

The dataset used for this analysis consists of the manual journal entries of a multinational company. Only the fiscal year 2015 is selected because this year is fully available.

Each manual journal entry posting has at least two records; a debit and a credit record. Both records have their own account on which the record is posted. An example is given in Table 14: A bank payment is made to buy office supplies for 5.000 euro. The balance on the bank account is credited, so there is 5.000 deducted from the balance on the bank. These costs are debited on the office supplies account. In this way the payment is recorded in the right way.

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Line item nr.</th>
<th>Account name</th>
<th>Debit/Credit</th>
<th>Amount</th>
<th>User</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>10001</td>
<td>1</td>
<td>Bank account</td>
<td>Credit</td>
<td>5.000,-</td>
<td>USERX</td>
<td>...</td>
</tr>
<tr>
<td>10001</td>
<td>2</td>
<td>Office supplies</td>
<td>Debit</td>
<td>5.000,-</td>
<td>USERX</td>
<td>...</td>
</tr>
</tbody>
</table>

The auditor will expect certain accounts to contain only debit or only credit transactions. An example is given in Table 14. The account ‘office supplies’ is an expense account and will usually only consist of debit postings. An overview of the type of accounts to which is normally only debited or credited is shown in Appendix M.

The dataset consist of 4.134 records. The relevant features for this research are selected; ‘G/L account name’ and ‘Debit/Credit indicator’. For a unique ID for each transaction the ‘transaction ID’ and ‘line item number’ are combined in one feature.

Model description

The data is imported in Rapidminer. A model is built in Rapidminer to apply the k-NN anomaly detection algorithm. The results are shown in the next paragraph.
Results
The results of the model are shown in Figure 25 and Table 15. The number of postings to one account differ between 14 and 604 records. The number of anomalies are in total 12 postings. The number of anomalies per account differ between 1 and 7. Most typical ‘debit’ accounts will not contain a credit posting. These are accounts A and H. Typical ‘credit’ accounts will not contain a debit posting. This is in this case account D. Accounts B, C, E, F and G contain exceptions. One in account B, C and G, two in account F and 7 in account E.

<table>
<thead>
<tr>
<th>OUTPUT</th>
<th>K-NN OUTLIER SCORE</th>
<th>DEBIT/CREDIT INDICATOR</th>
<th>G/L ACCOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>1</td>
<td>Debit</td>
<td>Account B</td>
</tr>
<tr>
<td>#2</td>
<td>1</td>
<td>Debit</td>
<td>Account C</td>
</tr>
<tr>
<td>#3</td>
<td>1</td>
<td>Credit</td>
<td>Account G</td>
</tr>
<tr>
<td>#4</td>
<td>0.92</td>
<td>Credit</td>
<td>Account F</td>
</tr>
<tr>
<td>#5</td>
<td>0.92</td>
<td>Credit</td>
<td>Account F</td>
</tr>
<tr>
<td>#6</td>
<td>0.50</td>
<td>Credit</td>
<td>Account E</td>
</tr>
<tr>
<td>#7</td>
<td>0.50</td>
<td>Credit</td>
<td>Account E</td>
</tr>
<tr>
<td>#8</td>
<td>0.50</td>
<td>Credit</td>
<td>Account E</td>
</tr>
<tr>
<td>#9</td>
<td>0.50</td>
<td>Credit</td>
<td>Account E</td>
</tr>
<tr>
<td>#10</td>
<td>0.50</td>
<td>Credit</td>
<td>Account E</td>
</tr>
<tr>
<td>#11</td>
<td>0.50</td>
<td>Credit</td>
<td>Account E</td>
</tr>
<tr>
<td>#12</td>
<td>0.50</td>
<td>Credit</td>
<td>Account E</td>
</tr>
</tbody>
</table>

Debit/Credit indicators

Discussion
As the results in Figure 25 show, the k-NN anomaly detection algorithm is very well able to detect the anomalies that are in the dataset. The groups of postings to a specific account number that only consist of either debit postings or only consists of credits postings have an anomaly score assigned equal to zero. Only the deviating postings get an anomaly score above 0. There is a difference in anomaly score between the accounts with one, two or seven anomalies. It makes sense that the anomaly score decreases when the number of anomalies per account increases.

So, in practice, the k-NN algorithm could be applied to scan different accounts for unusual debit or credit postings when auditing the financial reporting process. In addition, this method could be used to detect deviations in (almost) real time by implementing it in a control tool. In practice, every company can have their own mapping of accounts and account names in their information systems. The proposed debit/credit check is more flexible than making a strict control on every account. This is also very useful for external audit firms since they review multiple clients, each with their own controls on their IT environment. By using the anomaly detection algorithm this could make the audit process more flexible and data-driven.

The current model investigates eight different accounts. The auditor can decide which accounts are relevant for every situation. The number of accounts that is included can be extended if more accounts are interesting.
The model is tested to evaluate manual journal entry postings in a fast and flexible way. It should be noted that anomaly detection algorithms perform well only if there are big groups of the same data points and relative few exceptions. In the manual journal entries there are a lot of accounts to which is only posted a few times (i.e. total postings less than 20). The k-NN anomaly detection algorithm will classify these small groups also as anomalies. Therefore, the anomaly detection algorithm should not be used to detect exceptions in small amounts of data.

Deployment
The analysis is triggered by an auditor that investigates the manual journal entries of a company. The process of anomaly detection is described step by step and is shown in Figure 26.

First, the auditor should determine the typical ‘debit’ and ‘credit’ accounts based on their knowledge of the company and general audit knowledge. After the journal entry data is entered in the tool, only the relevant features are selected. In addition, the accounts of interest are selected and a threshold is set for the anomaly score of the output. For the k-NN algorithm an anomaly score close to 0 will be a normal case and an anomaly score close to 1 will be an exception. The optimal threshold will depend on the dataset and the desired output of the domain expert. The model will be applied and the transactions in the output can be investigated by the auditor. If needed, the threshold for the anomaly score can be adjusted to influence the number of results.

Figure 26: Anomaly detection process
5.3.2 Detection of tax code anomalies in purchase transactions

The two dimensional anomaly detection problem in the previous paragraph showed interesting results. The ‘debit/credit indicator’ in that test was a variable with only two possible values (i.e. binary). It would be interesting if the anomaly detection algorithms still perform well when the problem gets more complex. A more complex anomaly detection model is identified based on an interview conducted with an IT auditor from a public audit firm (Appendix N).

When a company purchases goods they automatically pay value added taxes (VAT) on the products they buy. However, unlike consumers, companies do not have to pay these taxes. VAT should only be included on the final product that is bought to consume. Therefore a company assigns a tax code to every purchase transaction in the ERP-system. This tax code corresponds to a certain percentage and type of tax that links to the purchase. The different tax codes exist because of different tax rates in the countries of origin of the product. Also within a country, the tax on different products can vary. Assigning the right tax code to a transaction can go wrong in practice and can result in problems for a company. They can be exposed to a ‘tax risk’ when claiming more VAT than they deserved or miss a ‘tax chance’ when less VAT is claimed than deserved.

What happens in practice is that some tax codes are used very often for a specific supplier over time. This makes sense because the same kind of products will usually be bought from the same supplier. Anomaly detection can be of added value here to detect tax codes that are in general often used, however, are very rarely occurring at a specific supplier.

The features used in this analysis are the following: A business has multiple suppliers (feature: Vendor) from which is purchased for a specific value (feature: Value). To each purchase order a tax code (feature: Tax code) is assigned. For extra analysis also the feature ‘date’ and ‘value’ are selected from the initial dataset.

Data description

For this analysis a dataset from a multinational company is used. The dataset consists of information regarding purchase transactions. First, the relevant features are selected:

- Transaction ID (only used as ID)
- Vendor nr.
- Tax code

The purchase transactions took place during the first six months of 2015. 50 Different suppliers are present in the dataset. There are 8 different tax codes. For this analysis, the 14 suppliers with the most transactions are selected (number of transactions > 20) because the anomaly detection algorithms do not perform well on small amounts of data. The anomaly detection algorithm would indicate a whole set of transactions of one supplier as anomalies. The resulting dataset used for analysis consists of 18,820 records. No data transformation or normalization is needed because both variables are categorical.

Model description

The purchase transaction data is imported in Rapidminer. A model is built to apply the k-NN anomaly detection algorithm and the output is shown in the next paragraph.
Results

The output of the anomaly detection model for exceptional tax anomalies is shown in Table 16. The total number of transactions is 18,820. There are 13 exceptions found with an anomaly score threshold of 0.9.

The visualization of the results is shown in Figure 27. On the x-axis the different tax codes are set out. On the y-axis the different vendors are presented. A point in the figure represent a single transaction and the color shows the outlier score. The big groups of blue dots represent the bulk of the transactions. The blue color shows these transactions are considered normal. There are also various smaller groups of transactions for each vendor. These show the transactions with tax codes that are rarely used by a vendor.

Table 16: Tax code anomaly detection output

<table>
<thead>
<tr>
<th>TRANSACTION KEY</th>
<th>OUTLIER SCORE*</th>
<th>TAX CODE</th>
<th>VENDOR NR.</th>
</tr>
</thead>
<tbody>
<tr>
<td>777001X3</td>
<td>1,000</td>
<td>100TAXCH X3CH</td>
<td>1000000600239</td>
</tr>
<tr>
<td>000002V0</td>
<td>1,000</td>
<td>100TAXCH V0CH</td>
<td>1000000600106</td>
</tr>
<tr>
<td>000002X1</td>
<td>1,000</td>
<td>100TAXCH X1CH</td>
<td>1000000600142</td>
</tr>
<tr>
<td>817001X1</td>
<td>1,000</td>
<td>100TAXCH X1CH</td>
<td>1000000600012</td>
</tr>
<tr>
<td>596001C1</td>
<td>1,000</td>
<td>100TAXCH C1CH</td>
<td>1000000600082</td>
</tr>
<tr>
<td>146001V0</td>
<td>1,000</td>
<td>100TAXCH V0CH</td>
<td>1000000600320</td>
</tr>
<tr>
<td>547001X3</td>
<td>0,950</td>
<td>100TAXCH X3CH</td>
<td>1000000600082</td>
</tr>
<tr>
<td>589001X1</td>
<td>0,950</td>
<td>100TAXCH X1CH</td>
<td>1000000600046</td>
</tr>
<tr>
<td>869001X3</td>
<td>0,950</td>
<td>100TAXCH X3CH</td>
<td>1000000600082</td>
</tr>
<tr>
<td>294001X1</td>
<td>0,950</td>
<td>100TAXCH X1CH</td>
<td>1000000600046</td>
</tr>
<tr>
<td>408001X1</td>
<td>0,900</td>
<td>100TAXCH X1CH</td>
<td>1000000600153</td>
</tr>
<tr>
<td>000007X1</td>
<td>0,900</td>
<td>100TAXCH X1CH</td>
<td>1000000600153</td>
</tr>
<tr>
<td>000001X1</td>
<td>0,900</td>
<td>100TAXCH X1CH</td>
<td>1000000600153</td>
</tr>
</tbody>
</table>
*(OUTLIER SCORE THRESHOLD = 0.9)

Figure 27: Visualization tax anomalies
Discussion of the results
The presentation in Figure 27 provides an overview of the anomaly detection model performance. There are tax codes that are used very often, but only once for a specific supplier. These are the most interesting transactions for the analyst and they are detected very well. These transactions are colored red which means they get the highest anomaly score by the anomaly detection algorithm. The output is also discussed with an domain expert (Appendix O) and he confirmed that these infrequent occurring tax codes are relevant to investigate in practice.

The current approach to find tax code anomalies is done by ‘targeted analytics’ (i.e. specific queries) (Appendix O). In the current setup, transactions with a tax code that appears in less than 5% of the transactions (when the data is categorized per supplier) are selected. There is assumed that these 5% buckets contain a wrong tax code. Dependent on the number of transactions, this 5% bucket can be too small or too large. The k-NN anomaly detection algorithm is in this way of added value since it is adaptable to the dataset that is analyzed. It does not rely on a specific percentage and is therefore helpful when the optimal percentages are not known and when they are different per supplier.

Point 1 in Figure 27 shows a transaction of which the tax code is only used once for that supplier. The two transactions indicated with ‘2’ in Figure 27 show a tax code that is used two times and the three transactions at ‘3’ show a tax code that is used only three times for one supplier. The transactions in Table 16 provide an overview of the transactions with an outlier score of 0.9 or higher. This output includes exactly all tax codes that occur once, two, or three times per supplier. Based on the desired output, the threshold for the anomaly score can be adapted such that transactions with tax codes occurring in bigger groups will be shown.

The model is also tested with three parameters. The extra parameter was the date on which the transaction took place. In practice it could be that one tax code is used only during a certain period and by exception on another period in a year. However, a major drawback when using three dimensions (tax code, vendor, date) is that anomalies are found in every combination: tax code vs. vendor, tax code vs. date and date vs. vendor. The combination ‘date vs. vendor’ does not have any meaning though for the detection of tax anomalies. Overall, including a third feature does lead to noisy results and the anomaly detection model will focus more on outlying values in one feature.

Robustness of the k-NN algorithm: Transformation from continuous to categorical variables
Based on the results in this chapter can be concluded that the k-NN algorithm is capable of detecting anomalies in different situations. However, the k-NN algorithm can perform less than expected when some extreme values are present in the data. In this paragraph is shown how a data transformation step can be included to get the expected results from the anomaly detection algorithm.

The problem that shows up can be described by the following example. Specific tax codes are usually used in a specific price range (Figure 28). However, some exceptions occur in the data with an extreme high price for that tax code (point 1 to 8 in Figure 28). Due to the fact that point 1 is present in the data, the anomaly detection algorithm is not able to distinguish point 7 and 8 very well from the other data points. This follows from the red color of point 1 (indicating a high anomaly score) and the blue color of point 7 and 8 (indicating a low anomaly score). So the robustness of an anomaly detection model is limited when using continuous variables.
This issue can be solved by a data transformation step before the k-NN anomaly detection algorithm is applied. In Figure 29 and 30, an example is provided of such a transformation of the input data to let the anomaly detection algorithm generate the expected output. The 8 different tax codes are plotted on the x-axis and the date of the transaction is set out on the y-axis. Point 1 is highlighted in both figures. This transaction has a tax code that is often used, however, not in that time of the year. Therefore point 1 is an interesting exception to be detected.

The date in Figure 29 is represented by an integer value and afterwards normalized. Point 1 in the analysis of Figure 29 is not detected as an anomaly though (blue color represents a low anomaly score). To change this, the feature date can be aggregated to groups per month. The results of the anomaly detection model after this data transformation are shown in Figure 30. Each transaction in a group with a lot of transactions gets a low anomaly score. In Figure 30, point 1 is detected as an anomaly since it is the only transaction in that period for that company.

Conclusion

Automatically detecting transactions with an anomalous tax code is very well possible with the k-NN anomaly detection algorithm. Especially problems with two categorical variables are well suited to be analyzed with the k-NN anomaly detection algorithm since they are more robust to fluctuations in the data. Data transformation steps can be used to provide the right input to an anomaly detection algorithm and find the expected transactions in the output. This is shown by transforming the ‘date’ value into buckets per month. Specific to each problem type, different aggregation and transformation steps can be required.
**Deployment**
An overview is made of the process to use the ‘tax anomaly detection model’. The process is shown in Figure 31 and described shortly below.

The process is triggered by an client that wants to know if there might be anomalous tax codes in a set of purchase transactions. After loading the data in the tool, the two features of interest are selected. Afterwards the suppliers of interest are selected and the anomaly detection model will provide the transactions with a deviating tax code. The threshold can be adjusted to influence the number of results.

![Anomaly detection process 'tax anomalies'](Figure 31: Anomaly detection process 'tax anomalies')

---

**5.4 Include feedback to improve an anomaly detection algorithm output and create learning effect**

An anomaly detection algorithm learns based on the intrinsic information in the dataset to find anomalies. There is however no learning effect in an unsupervised machine learning algorithm. So, each new dataset is approached as an independent dataset. It is however very well possible that an anomaly detection algorithm is applied on multiple datasets with the same properties. For example, if the same data is analyzed for a company every quarter of a year and the engagement covers a period of ten year, this will already lead to the same test being executed 40 times. To make the audit process more efficient, false positive results out of an anomaly detection model should be limited as much as possible. It would therefore be beneficial if the anomaly detection algorithm also could learn from previous experience with some user input.

The nature of the machine learning problem changes from unsupervised to supervised when providing user input. Table 17 provides a comparison of important characteristics and assumptions of both anomaly detection algorithms and supervised learning algorithms (Ng, 2016).

*Table 17: Comparison of main characteristics anomaly detection vs. supervised learning*

<table>
<thead>
<tr>
<th></th>
<th>Anomaly detection</th>
<th>Supervised learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number and distribution of positive/negative examples.</strong></td>
<td>Very small number of positive examples (anomalies) and large number of negative examples (normal data points).</td>
<td>Large number of both positive and negative examples.</td>
</tr>
<tr>
<td><strong>Dynamic nature of anomalies</strong></td>
<td>Many different types of anomalies and future anomalies may be different from what we’ve seen so far.</td>
<td>Future examples are similar to the ones in the training set.</td>
</tr>
<tr>
<td><strong>Label availability</strong></td>
<td>No prior information about the examples is needed as input.</td>
<td>Classification needed about which examples are ‘normal’ and which are ‘anomalous’.</td>
</tr>
</tbody>
</table>
The number of anomalies in a dataset is by definition limited. It would therefore be very hard to train a machine learning model in order to classify the output of the anomaly detection algorithm with such a model. The training set would be too small. In addition, to be able to train such a model the data should be labeled which could be a time consuming process if auditors should do it. Finally, future examples of anomalies should already have occurred in the test set to be able to predict them. Therefore can be concluded that the combination of anomaly detection with supervised learning would not be advantageous.

Two alternatives have more potential. In some specific cases the problem can be approached as a supervised learning problem from the start. An example for transaction level data is the detection of fraudulent credit card transactions. Predictive models for credit card fraud detection are already in active use in practice (Bhattacharyya, Jha, Tharakunnel & Westland, 2010). With a supervised approach, the machine learning model can be learned what normal behavior is and therefore the model is less sensitive to generate false positive results compared to anomaly detection algorithms which do not include this information. A supervised approach requires totally different algorithms and a different setup and is therefore not further discussed in this study.

The second alternative is to combine anomaly detection algorithms with programmed rules in order to prevent false positive results. For example, in the tax anomaly detection model in 5.3.2 you can generate rules that limit the output. Three examples are given here.

- Include a rule that you are not interested in a specific tax code at all (Points in region 1, Figure 33, are removed).
- A rule that you are not interested in a certain combination of supplier and tax code (Points in region 2, Figure 33, are removed).
- A rule that you are only interested in a certain combination of supplier and tax code if the anomaly score is higher than X (Points in region 3, Figure 33, are removed).

By combining unsupervised machine learning in a smart way within current rules and queries the flexibility and adaptability of different audit tests can increase. In addition, by adopting unsupervised anomaly detection in advisory tools, the tools can adapt to different situations.
Chapter 6: Conclusions, limitations & further research

In this chapter conclusions of this study will be drawn. Afterwards, the limitations will be discussed and areas for further research will be indicated.

6.1 Conclusions

In this study is investigated how machine learning and more specifically, unsupervised anomaly detection algorithms, can be applied to detect exceptions in transaction level data. The study gives innovating insights in the application of anomaly detection in a new domain and provides the groundwork for further studies in this field. In this section the main results will be discussed for each part of this study.

Current data analysis in audit

In the current situation, data examination is mostly done by IT auditors and the most used type of analysis is ‘data examination and query’. However, general auditors have the knowledge about regulations but make mostly use of simple spreadsheet software Excel. More sophisticated tools standard anomaly detection algorithms can improve their effectiveness by guiding them to the most deviating transactions in datasets. Rapidminer models are intuitive and easy to use which makes more advanced algorithms available to general auditors. In addition, the visualizations of Rapidminer provide added value for auditors in their day to day work.

Anomaly detection algorithms

Unsupervised anomaly detection is a realistic and widely applicable type of machine learning for the detection of exceptions in transaction level data because only unlabeled data is needed. Exceptions are infrequently occurring data points which can be detected automatically with an anomaly detection algorithm. The dynamic and unpredictable nature of exceptions contributes to the suitability of the technique too. The algorithms are distance based so they do not make assumptions about the underlying data and multiple features can be compared.

A lot of different anomaly detection algorithms exist in literature. There can be concluded that for the detection of anomalies in transaction level data in our study, the k-Nearest Neighbor algorithm is the most promising algorithm. This is in line with research of, for example, Amer et al. (2013). The strength of the k-NN algorithm stems from the fact that it is inherently unsupervised and has intuitive criteria for the detection of anomalies. The k-NN algorithm performs well in detecting global anomalies and is least sensitive to local anomalies.

The sensitivity of the k-NN algorithm and the one-class SVM algorithm to different parameter values is very limited. The uCBLOF algorithm is more sensitive to different values of k. This result is in line with Goldstein & Uchida (2016) which score the k-NN, one-class SVM and uCBLOF respectively ‘+’, ‘+’ and ‘o’ on sensitivity to parameter values. The low sensitivity contributes to the appropriateness of the k-NN anomaly detection algorithm because the optimal parameter value is often hard to determine in advance.

In this study the computation time of the k-NN algorithm was also limited (some seconds in all models) compared to the time it takes to cluster the data for the uCBLOF algorithm (about 30 minutes in the model of chapter 4) and computation time of the one-class SVM algorithm (about 1 minute in the model of chapter 4).
Point anomaly detection model

Proper input data is very important to successfully apply an anomaly detection algorithm. All data should be normalized and data transformation or aggregation may be needed to get the expected output from the algorithm. This can for example be done by changing a feature ‘date’ to a categorical variable. It is also important to include only the parameters of interest. Irrelevant parameters will mess up the results instead of contributing to a pattern.

When considering multiple features (i.e. more than 2) together in an anomaly detection model, the anomaly detection algorithm mostly detects extreme values (for continuous variables) and infrequent occurring values (for categorical variables). This is shown to be useful when identifying exceptional sales transactions. Extreme values in for example value, cost and profit and infrequent occurring entry users can be automatically identified. This application is adaptive to each situation and does not rely on any specific rules. Due to the anomaly score that is assigned to each transaction the model output guides the auditor to the most deviating transactions first. In this way anomaly detection provides an easy way to make the work of auditors more data driven.

The anomaly detection model can be applied in different situations of the audit process. In the planning phase to identify risk areas, during the interim control for detection of process deviations and during the year end control to identify suspicious transactions. Apart from sales transactions, other types of transactions may also have the potential to benefit from a quick scan with an anomaly detection model; purchasing transactions, bank receivables/payables and manual journal entries are suggested to be analyzed for extreme or infrequent occurring values.

Contextual anomaly detection models

Anomaly detection can detect contextual anomalies when considering two features together. A contextual anomaly is a transaction that is only anomalous in a specific context but not otherwise. These values will not be found when using outlier detection in one feature only.

Four different situations in the audit environment are shown in which this technique can be applied. Based on synthetic data, anomaly detection can detect invoices that have a relative long period of payment outstanding. Second, the k-NN algorithm can discover payments to a wrong supplier when the amount differs from the norm. Based on real world data, the k-NN algorithm is able to detect credit postings to debit accounts and vice versa. Finally, the algorithm can automatically find tax code anomalies that are unusual for a specific supplier.

In the contextual anomaly detection applications, the k-NN algorithm performs in particular well for the detection of anomalies in categorical variables. The application on deviating ‘debit/credit’ postings as well as the application on tax code anomalies showed good results. However, when applying anomaly detection to continuous variables, some extreme values will have a big influence on the results. Contextual anomalies are then hardly distinguished from normal cases. The k-NN anomaly detection algorithm shows more robust results when applied to categorical variables.

Contextual anomaly detection can be adopted in audit and advisory tools when combining the adaptive algorithms in a smart way with hard coded rules and queries. There should however be noted that anomaly detection is only based on the intrinsic properties of the data and only detects rare and deviating data points. The exceptions that are interesting should have these properties to be detected. Because no external information is provided to the algorithm it can generate false positive results if other, non-interesting, transactions have the same character; being rare and deviating from the norm.

Overall can be concluded that anomaly detection has potential to make audit tests more flexible and adaptive in specific situations. There has however been taken care of the limitations of the algorithms.
6.2 Limitations and further research

This study has given insight in the application of anomaly detection in the audit domain. The study also has some limitations which are discussed in this section. Areas for further research are provided that follow from these limitations.

External learning effect for anomaly detection algorithms

There is investigated how to improve anomaly detection algorithms to learn not only based on intrinsic properties of the data but also from the interpretation of the output. This results however in a combination of supervised machine learning and anomaly detection which is not recommended. It has more potential to limit false positives with coded rules or to approach a problem as a supervised machine learning problem from the start.

Supervised learning

In this study is focused on unsupervised learning because it is realistic in practice due to unavailability of labels. The anomaly detection algorithms are not good in identifying a link between multiple features. Supervised learning algorithms have shown better results in this task in other studies. The difference between anomaly detection and supervised learning can best be shown by an example; Take supplier X to which every month 100 euro is transferred and suddenly 1000 euro. Anomaly detection will detect this transaction because the value 1000 is occurring very infrequent for that specific supplier. Supervised machine learning could detect this value because he has learned from an initial test set that the combination ‘supplier X’ and ‘100 euro’ is normal.

Supervised learning models are predictive and able to update based on historical data. By learning what normal transactions look like and how anomalous transactions look like it could be possible to make a predictive model. It is however needed to have a big enough test set with examples of both normal and anomalous behavior to train a model. An auditor should label every single transaction to indicate if he considers it normal or anomalous. In addition, the implicit assumption is made that normal and anomalous behavior is consistent over time and the test set includes examples of anomalies that will occur in the future. There should be investigated in which specific scenarios it is possible to make such a model. Fraudulent credit card transactions and fraudulent insurance claims are examples of successful predictive models based on transaction level data. Specific audit knowledge is recommended to understand which aspects are important to consider.

Additional meaningful features

In this research, several features are identified that could indicate meaningful exceptions. A more in depth analysis can be done to investigate if there are additional variables that could indicate interesting deviations for auditors.

Every transaction has a text field that describes what the transaction is about. This textual information could contain meaningful information for the detection of exceptions in transaction level data. Including text mining in a tool could be further investigated.

Anomalies in analyzed dataset

Information about whether the datasets contain errors was not available (i.e. the available data has no labels). In the first model (chapter 4) is therefore assumed that values outside specific boundaries are exceptions. These boundaries may be company and situation specific and in each situation different anomalies might show up. Therefore, the performance of the algorithms may differ in different situations. It could be helpful for further research to capture datasets that contain high risk transactions or record the exceptions that are found by auditors. This information would be useful to be studied.
Research in cooperation with engagement team
This research focusses mainly on the way to automatically find anomalous data points in datasets with the use of anomaly detection algorithms. There is however less attention paid to the real meaning of the anomalies within the business environment, which could be interesting too. This is done on purpose because it cannot be known in advance if a deviating transaction is indeed an error. The real meaning of an anomaly is very dependent on the specific engagement and situation in which the anomaly occurs and requires audit knowledge specific to that situation. It would lead to more case specific and audit related information while this study focusses on the information systems aspects. In addition, the data used in the different tests is derived from real client data from which the financial figures are already approved by the auditors. If there are transactions found and discussed with the responsible audit team that are not correct, the audit company could be at risk. Therefore is chosen not to discuss the model with auditors of the specific engagement team but with other non-related auditors. To overcome this limitation, future research could perform analysis of data in the same period as the actual audit of the company. This makes it possible to reflect very precise on the exceptions that are found by a new method and compare them to what is found by regular analysis along audit templates.

More advanced anomaly detection models
In this study is focused on different applications of anomaly detection to assist auditors in quick analysis of datasets. Anomaly detection could be adopted in more advanced models. Further research could for example make use of programming in R.

Qualitative research
During this research much qualitative evaluation is used. For example, different interviews are conducted with domain experts in the company. Different respondents may have led to slightly different results. More respondents could have been assessed to make the outcomes of the interviews more valid. Due to the ‘busy season’ at the end of this project it was harder to make appointments with auditors since a lot of them were working at the location of the client.

This research focused on the identification of different situations and tests to apply anomaly detection. Different limitations arise from the exploratory character of this study and the different case studies that are selected. The selection of the chosen applications may be subjective. Another researcher in another situation may have chosen other applications to show how anomaly detection performs. In addition, it is hard to generalize the results and to come up with very precise results. By analyzing different tests is tried to capture a more general picture on the performance of the different algorithms to improve the generalizability of the results.
References


Bay, S., Kumaraswamy, K., Anderle, M. G., Kumar, R., & Steier, D. M. (2006). Large scale detection of irregularities in accounting data. In Sixth International Conference on Data Mining (ICDM'06) (pp. 75-86). IEEE.


Tan, P.N., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining. Addison-Wesley.


Appendices

Appendix A: Gartner Hype Cycle of Technology 2016

Figure 34: Gartner Hype Cycle of Technology
Appendix B: Summary of unstructured interviews conducted to find appropriate transaction type for the application of anomaly detection.

Identification of application area’s for machine learning

Unstructured interviews to identify interesting area’s to apply unsupervised machine learning on transaction level data in the audit environment.

Goal:

The goal of this research is to explore possible applications of unsupervised machine learning in transaction level data to find exceptions interesting to the auditor. Therefore, first three unstructured interviews are conducted to get insights in what aspects are important for auditors. Also is investigated where room for improvement may be. In this way will be tried to get more insight and feeling with the audit data. In addition, the main goal of these interviews is to find a type of transactions that is interesting to investigate further and build an anomaly detection model for.

Interviewees:

Name: Dennis van de Wiel  
Function: Senior Manager ITA  
Date: 27-05-2016

Name: Sander Kuilman  
Function: Manager ITA  
Date: 01-06-2016 / 06-06-2016

Name: Bert Scherrenburg  
Function: Senior Manager ITA  
Date: 01-06-2016

Conclusions:

Different processes could be analyzed. For example, the purchasing process. However, since the purchasing process contains outgoing payments there is already a lot of control in this area and therefore there may be little added value. Another area that is mentioned is the manual journal entry process. However, this is a process with very many different kind of transactions. Unsupervised learning and anomaly detection is better in detecting a small deviations from a large number of more or less the same transactions. Finally, the sales process is mentioned to be interesting to investigate for a new type of analysis. The sales transactions are chosen because the characteristics (large number of same kind of transactions) are well suited to apply anomaly detection on.
Appendix C: Description of selected features of the sales transaction data

Table 18 shows the 9 selected attributes from the dataset. No missing values were detected. ‘Cost of goods sold’ represent the costs of making the product. ‘Revenue’ is the sales value of the product. ‘Profit’ is equal to the difference between revenue and cost of goods sold. The quantity of the product sold is described by two values; ‘quantity ordered’ and ‘quantity delivered’. Every order has a date on which it is created, the ‘document date’. The timing of entering the order in the system is described by two features; ‘entry time’ and ‘entry date’ referring to the time of the day and the date of entering the order in the system.

**Table 18: Selected features from dataset**

<table>
<thead>
<tr>
<th>#</th>
<th>Attribute</th>
<th>Attribute type</th>
<th>Reason to select</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cost of goods sold</td>
<td>Continuous</td>
<td>Large and infrequent transactions are interesting to investigate more throughout</td>
<td>- Boynton &amp; Johnson (2006)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Argyrou (2013)</td>
</tr>
<tr>
<td>2</td>
<td>Revenue</td>
<td>Continuous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Profit</td>
<td>Continuous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Entry time</td>
<td>Interval</td>
<td>Transactions posted on odd times of the day</td>
<td>- CAQ (2008)</td>
</tr>
<tr>
<td>5</td>
<td>Entry date</td>
<td>Interval</td>
<td>Long lapse of days between original document date and entry date</td>
<td>- CAQ (2008)</td>
</tr>
<tr>
<td>6</td>
<td>Document date</td>
<td>Interval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Ordered quantity</td>
<td>Continuous</td>
<td>Transactions in which much more is delivered than ordered are interesting to</td>
<td>- Markham (2015)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>investigate (channel stuffing).</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Delivered quantity</td>
<td>Continuous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Entry user</td>
<td>Categorical</td>
<td>Entered by infrequent</td>
<td>- CAQ (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From the original features, three new features are created:

- Profit percentage
- Document date – Entry Date
- Delivery Surplus

They are described next.

**Profit percentage:**

The profit percentage is created based on the cost of goods sold as fraction of the revenue.

\[
\left( \frac{\text{cost of goods sold}}{\text{revenue}} - 1 \right) \cdot 100
\]

Since the revenue should be fixed or determinable (Johnstone et al., 2014), orders with exceptional values for the percentage of profit are interesting to the auditor.

**Document date and entry date difference:**

The difference between the document date and the entry date is calculated.

\[
\text{document date} - \text{entry date}
\]
In this way a integer variable is created that describes the delay or postponement of the recording of a sales order. This timing difference could influence the booking period the sales order is recorded in. Therefore, it is interesting when the value of this feature deviates from 0. The higher the difference, the more interesting it is.

**Delivery surplus:**

The delivery surplus variable is created based on the ordered quantity and the delivered quantity.

\[ \text{max}(\text{quantity delivered} - \text{quantity ordered}, 0) \]

When less is delivered than ordered this is not interesting to an auditor. This can happen due to normal business operations when there is not enough stock or production capacity to deliver on time. However, when there is much more delivered than ordered this is unusual. Markham (2015) define channel stuffing as a way to manipulate accounts and manage earnings by delivering more goods than needed. Usually there is a right of return which means the goods would be returned on a later date. In this way the sales are only temporally inflated and change the sales figures of the company.
Appendix D: Summary semi-structured interviews with audit managers

To evaluate the anomaly detection model there are five semi-structured interviews conducted with (general) auditors and audit managers. The interviews took approximately 30 minutes each of which 10 minutes was used to describe and show the model and 20 minutes were used to discuss different aspects of the model based on a questionnaire. Each auditor has written down their main message to the different questions of the questionnaire. These answers are shown per interviewee in this appendix. The answers are given in Dutch because this was more familiar for the auditors.

Name: Tim v. Tetrode
Function: Senior Manager Audit
Location: Eindhoven
Date: 31-08-2016

After you have seen the model(s), do you think this application of unsupervised anomaly detection is useful in practice, when auditing the sales environment?

*Ja, het lijkt me nuttig om het als additionele tool te gebruiken.*

Do you think the variables that are included in the anomaly detection model(s) are relevant? (If so, which are most interesting and why? Or why not?)

*Ja, ik beschouw ze allemaal relevant. Met name omzet, gebruiker en tijdstip is interessant. De cost of sales en profit vind ik het minst interessant omdat dit diverse oorzaken kan hebben (productie type met hoge/lage marge).*

In which stage of the audit process could this type of analysis (unsupervised anomaly detection) be valuable?

*Ja, dit is met name relevant voor de balanscontrole.*

Are the anomalies that are found by the model(s) interesting to investigate further? If so, which cases do you think are most relevant?

*Ja, met name de outliers die in meerdere buckets vallen zijn zeker interessant om verder op in te gaan.*

Do you think application of unsupervised anomaly detection is helpful in audit in general? If so, do you see other application area’s for unsupervised anomaly detection in audit?

*Ja, andere gebieden kunnen zijn:*
  - Manual journal entries
  - Betalingen in bankapplicatie
  - Inkopen
After you have seen the model(s), do you think this application of unsupervised anomaly detection is useful in practice, when auditing the sales environment?

*Naar mijn mening is het een efficiente manier op zowel tijdens interim als jaareinde controles snel zicht te hebben op volledige datasets (downloads) en risico’s in verschillende variabelen te identificeren. Dit programma is een interessante aanvulling op de zogenaamde ‘pivot tables’ in Excel, omdat naast de samenvatting van het bestand ook outliers worden geïdentificeerd.*

Do you think the variables that are included in the anomaly detection model(s) are relevant? (If so, which are most interesting and why? Or why not?).

*De variabelen zijn interessant over verschillende fases in de controle. Zo zijn we tijdens de interim geïnteresseerd in risico’s tijdens processen, zoals bijvoorbeeld functiescheiding en zou de variabele “gebruiker” erg van toepassing zijn. Tijdens de jaareindecontrole zijn we geïnteresseerd in gegevensgerichte verwerking van data en zou de variabele ‘profit margin’ van toepassing zijn op de marge-analyse.*

In which stage of the audit process could this type of analysis (unsupervised anomaly detection) be valuable?

*Zie bovenstaande.*

Are the anomalies that are found by the model(s) interesting to investigate further? If so, which cases do you think are most relevant?

*Outliers worden besproken met de klant en eventueel onderbouwd met aanvullende documentatie.*

Do you think application of unsupervised anomaly detection is helpful in audit in general? If so, do you see other application area’s for unsupervised anomaly detection in audit?

- Inkoop proces
- Journal entry testing
- Betalingsorganisatie
After you have seen the model(s), do you think this application of unsupervised anomaly detection is useful in practice, when auditing the sales environment?

*Ja, ik denk zeker heel bruikbaar om snel inzicht te krijgen in de afwijkingen van het normale process, waar dus ook de grootste risico’s zitten.*

Do you think the variables that are included in the anomaly detection model(s) are relevant? (If so, which are most interesting and why? Or why not?)

*Ja, dit zijn wel de afwijkingen waar wij als accountant bij onze eigen data-analyse ook naar zouden kijken.*

In which stage of the audit process could this type of analysis (unsupervised anomaly detection) be valuable?

*Bij de interim-controle ➔ bij t.a.v. process (welke personen voeren in, wat zijn hun autorisaties/bevoegdheden

*Bij de final ➔ t.a.v. afwijkingen in de data.*

Are the anomalies that are found by the model(s) interesting to investigate further? If so, which cases do you think are most relevant?

*Belangrijke ➔ hoge warden, marge, invoer door niet standard mensen.*

Do you think application of unsupervised anomaly detection is helpful in audit in general? If so, do you see other application area’s for unsupervised anomaly detection in audit?

*Analyse grootboek:*

- Debet boekingen op credit rekeningen
- Afwijkende tegenrekeningen
- Afwijkende dagboeken

*Betalingen*

*Tussenrekeningen*
After you have seen the model(s), do you think this application of unsupervised anomaly detection is useful in practice, when auditing the sales environment?

*Ja, geeft inzicht in organisatie / processen en helpt bijzonderheden in kaart te brengen (Bijzonderheden hoeven niet per se fout te zijn). Interessant dat je de afwijkingen vind zonder dat je een threshold hoeft aan te geven.*

Do you think the variables that are included in the anomaly detection model(s) are relevant? (If so, which are most interesting and why? Or why not?).

*Ja, dit zijn variabelen waar wij ook naar kijken. Wel afhankelijk van de business welke variabelen meer en minder van belang zijn. In bijvoorbeeld wholesale vs. Retail worden andere marges gehanteerd.*

In which stage of the audit process could this type of analysis (unsupervised anomaly detection) be valuable?

*Zowel bij de YE (year-end control) om bijzonderheden te identificeren als bij interim of eerder om kennis van de organisatie te verkrijgen.*

Are the anomalies that are found by the model(s) interesting to investigate further? If so, which cases do you think are most relevant?

*Marge → splitsen naar business/type
Leverdatum → achterliggende reden zoeken voor grote verschillen
Vervolg is outliers koppelen om te zien wie outliers ‘maakt’ en terug naar de klant om te bespreken.*

Do you think application of unsupervised anomaly detection is helpful in audit in general? If so, do you see other application area’s for unsupervised anomaly detection in audit?

*Ja, voor inzicht.
Ja, belangrijk is de link tussen processen.*
After you have seen the model(s), do you think this application of unsupervised anomaly detection is useful in practice, when auditing the sales environment?

*Ja, niet alleen voor de sales. Ook inkoop, betalingen, journal entries.*

Do you think the variables that are included in the anomaly detection model(s) are relevant? (If so, which are most interesting and why? Or why not?).

*Ja, moeten wel oppassen met vergelijkbare variabelen. Zoals hoge kosten en lage winst. Door beide op te nemen scoor je 2x op feitelijk hetzelfde punt.*

In which stage of the audit process could this type of analysis (unsupervised anomaly detection) be valuable?

*Aangezien een dergelijke analyse richtinggevend is aan de audit zou ik dit tijdens de planningsfase al willen betrekken. Tijdens de final zou ik dan alle data analyseren.*

Are the anomalies that are found by the model(s) interesting to investigate further? If so, which cases do you think are most relevant?

*Denk ik zeker, zijn bijzonderheden en daardoor relevant. Hoewel overigens niet altijd hoge risico posten te zijn. Hoge bedragen vallen intern vaak ook op en zijn dan aan interne beheersing onderhevig.*

Do you think application of unsupervised anomaly detection is helpful in audit in general? If so, do you see other application area’s for unsupervised anomaly detection in audit?

*Kan zeker richtinggevend zijn. Moeten wel blijven opletten dat ook de niet outliers juist zijn. Zou mooi zijn als we een soortgelijke analyse kunnen gebruiken voor controle van volledigheid.*
Appendix E: Descriptive statistics and plots of the values of the eight features used in the anomaly detection model for sales transactions

In this appendix, some descriptive statistics are provided about the dataset that is used for the point anomaly detection model in chapter 4 (Table 19).

Table 19: Descriptive statistics dataset

<table>
<thead>
<tr>
<th>Feature</th>
<th>Average</th>
<th>St. dev.</th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost of goods sold</td>
<td>26987</td>
<td>61190</td>
<td>0</td>
<td>2618589</td>
</tr>
<tr>
<td>Revenu</td>
<td>53403</td>
<td>691332</td>
<td>0</td>
<td>39213017</td>
</tr>
<tr>
<td>Profit</td>
<td>26395</td>
<td>688465</td>
<td>-812252</td>
<td>39213017</td>
</tr>
<tr>
<td>Profit %</td>
<td>14%</td>
<td>5,15</td>
<td>-91%</td>
<td>20665%</td>
</tr>
<tr>
<td>Entry time</td>
<td>125042</td>
<td>23545</td>
<td>55235</td>
<td>222142</td>
</tr>
<tr>
<td>Document date - entry date</td>
<td>0,06</td>
<td>3,91</td>
<td>0</td>
<td>304</td>
</tr>
<tr>
<td>Delivery surplus</td>
<td>943</td>
<td>27350</td>
<td>0</td>
<td>1300000</td>
</tr>
<tr>
<td>Entry user</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Appendix F: How to set the parameter of the anomaly detection algorithm

The parameters of the different unsupervised anomaly detection algorithms need to be set before the model is applied. The question is how to set these parameters in the right way? This will be discussed per algorithm:

- Parameter k in k-medoids clustering.
- Parameter k in k-nearest neighbor global anomaly score.
- Parameter nu in one-class SVM anomaly score.

It should be noted that the k in the k-medoids clustering algorithm (k = number of clusters) has a totally different meaning than the k in k-nearest neighbor (k = number of data instances in the neighborhood that are considered when classifying a data point). So they are discussed separately.

K-medoids clustering

To get the best results, the parameter k in the k-medoids should be set beforehand. The optimal value for k can be found using the ‘elbow method’. As described by Leskovec & Rajaraman (n.d.) you can calculate the average within cluster distance for different values of k. When you set out the average cluster distance against the number of k you will see an elbow shape graph. The values will start high, fall rapidly until the best k, then change very little (Figure 35).

![Average within cluster distance](image1)

This is tested in Rapidminer with the “performance” operator. This operator calculates the average within cluster distance after the k-mediods clustering algorithm is applied. The average within cluster distance is calculated for different values of k. The results are shown in Figure 36.

![Average within cluster distance](image2)
Figure 36 shows the line drops very fast until k=9. Between k=9 and k=20 the average within cluster distance still decreases. After k=20 the line is almost flat. It is believed that the algorithms produce better results when the number of clusters k is overestimated (Amer & Goldstein, 2012). Following the ‘elbow rule’ it is in this situation advised to set k equal to 20.

**k-Nearest Neighbor**

For the nearest neighbor algorithm, the value to choose for k is not trivial (Veksler, n.d.). K should be large enough such that the error rate is minimized. Choosing k too small will lead to noisy decision boundaries. In addition, the density estimation for the records might not be reliable. However, k should be small enough such that only nearby samples are included. If k is too large this will lead to over smoothed boundaries. A general rule of thumb is (Veksler, n.d.):

\[ k = \sqrt{N} = \sqrt{\text{number of records}} \]

Another rule of thumb is given by Goldstein & Uchida (2016):

\[ 10 < k < 50 \]

Unfortunately there is no technique to determine the optimal k like in the previous example with the k-medoids clustering. Beside the use of these rules of thumb, the end user can try different values and investigate which parameter setting provides the most satisfying results. In this study is shown that the algorithm is not very sensitive to variations in the parameter setting. This is in line with e.g. Goldstein & Uchida (2016).

**One-class SVM algorithm**

The parameter nu is an upper bound on the fraction of anomalies and a lower bound of the fraction of support vectors relative to the total number of examples (Amer, Goldstein & Abdennadher, 2016). For example, if nu is set to 0.05 the algorithm will classify at most 5% of the data points being anomalies and at least 5% of the data instances being support vectors. The parameter is by default set to 0.5, indicating a upper bound of anomalies of 50%. Since in anomaly detection the number of anomalies is expected to be way less the parameter can be set lower. However, the value is also a lower bound for the fraction of support vectors. The higher the complexity of the model, the more support vectors are needed. Overall should be concluded that the optimal setting of the parameter is not easily determinable in advance. Based on other studies (e.g. Schölkopf et al.,2001) is advised to set the parameter between 0.05 and 0.5. In this study is however shown that the algorithm is not very sensitive to variations in the parameter setting.
Appendix G: Data labeling procedure

In the dataset, a new column is generated. This column represents if a record is labeled ‘normal’ or ‘anomalous’. Normal records get a 0 and anomalous records get a 1 assigned. An anomaly is a transaction that has an exceptional value in 1 or more features.

In addition to the label for the transaction as a whole, each feature is also labeled. An example is given in Table 20. So when a transaction has an exceptional value only in feature 4, the value for ‘outlier feature 4’ is equal to 1 and the value of ‘overall anomaly label’ is equal to 1. The rest of the values are set to 0. This provides an indication of how well the algorithm detects the anomalies in the different features.

Table 20: Example of transaction labels used for evaluation

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Overall anomaly label</th>
<th>Outlier feature 1</th>
<th>Outlier feature 2</th>
<th>Outlier feature 3</th>
<th>Outlier feature 4</th>
<th>Outlier feature 5</th>
<th>Outlier feature 6</th>
<th>Outlier feature 7</th>
<th>Outlier feature 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>160832-10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Appendix H: Anomaly detection performance per feature

To evaluate how well the anomalies in each of the features are detected is analyzed what the true positive rate is for each of the features. This is done for the three algorithms. The results are shown in Table 21, 22 and 23. Again the true positive rate in the top 100 results is used. So, for each feature is calculated what percentage of the total number of anomalies is represented in the top 100 results.

So for example there are 15 anomalous values in the feature ‘delivery surplus’. 11 out of the 15 are detected by the k-NN (k=10) algorithm in the top 100 results. Therefore a score of \( \frac{11}{15} \times 100 = 73\% \) is shown in Table 21.

Table 21: True positive rate per feature - k-NN algorithm

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
<th>Value</th>
<th>Profit</th>
<th>Profit%</th>
<th>Delivery surplus</th>
<th>Entry user</th>
<th>Entry time</th>
<th>Document date-Entry date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of outliers</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>15</td>
<td>57</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>k=10</td>
<td>100%</td>
<td>100%</td>
<td>88%</td>
<td>100%</td>
<td>73%</td>
<td>100%</td>
<td>67%</td>
<td>100%</td>
</tr>
<tr>
<td>k=20</td>
<td>100%</td>
<td>100%</td>
<td>88%</td>
<td>86%</td>
<td>73%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>k=30</td>
<td>100%</td>
<td>100%</td>
<td>88%</td>
<td>71%</td>
<td>73%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>k=40</td>
<td>100%</td>
<td>100%</td>
<td>88%</td>
<td>57%</td>
<td>67%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>k=50</td>
<td>100%</td>
<td>100%</td>
<td>88%</td>
<td>57%</td>
<td>67%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 22: True positive rate per feature - one-class SVM

<table>
<thead>
<tr>
<th>One-class SVM</th>
<th>Cost</th>
<th>Value</th>
<th>Profit</th>
<th>Profit%</th>
<th>Delivery surplus</th>
<th>Entry user</th>
<th>Entry time</th>
<th>Document date-Entry date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of outliers</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>15</td>
<td>57</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>nu=0.05</td>
<td>63%</td>
<td>100%</td>
<td>75%</td>
<td>57%</td>
<td>53%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>nu=0.1</td>
<td>63%</td>
<td>100%</td>
<td>75%</td>
<td>57%</td>
<td>53%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>nu=0.2</td>
<td>88%</td>
<td>100%</td>
<td>75%</td>
<td>57%</td>
<td>47%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>nu=0.3</td>
<td>88%</td>
<td>100%</td>
<td>63%</td>
<td>57%</td>
<td>47%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>nu=0.5</td>
<td>63%</td>
<td>100%</td>
<td>63%</td>
<td>57%</td>
<td>47%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 23: True positive rate per feature - uCBLOF

<table>
<thead>
<tr>
<th>uCBLOF</th>
<th>Cost</th>
<th>Value</th>
<th>Profit</th>
<th>Profit%</th>
<th>Delivery surplus</th>
<th>Entry user</th>
<th>Entry time</th>
<th>Document date-Entry date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of outliers</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>15</td>
<td>57</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>nu=0.05</td>
<td>25%</td>
<td>100%</td>
<td>38%</td>
<td>57%</td>
<td>20%</td>
<td>100%</td>
<td>58%</td>
<td>50%</td>
</tr>
<tr>
<td>nu=0.1</td>
<td>38%</td>
<td>100%</td>
<td>50%</td>
<td>57%</td>
<td>27%</td>
<td>100%</td>
<td>58%</td>
<td>50%</td>
</tr>
<tr>
<td>nu=0.2</td>
<td>63%</td>
<td>100%</td>
<td>63%</td>
<td>57%</td>
<td>40%</td>
<td>100%</td>
<td>58%</td>
<td>50%</td>
</tr>
<tr>
<td>nu=0.3</td>
<td>100%</td>
<td>100%</td>
<td>63%</td>
<td>57%</td>
<td>47%</td>
<td>100%</td>
<td>75%</td>
<td>100%</td>
</tr>
<tr>
<td>nu=0.5</td>
<td>38%</td>
<td>100%</td>
<td>50%</td>
<td>57%</td>
<td>27%</td>
<td>100%</td>
<td>58%</td>
<td>50%</td>
</tr>
</tbody>
</table>
Appendix I: Top 10 outliers per algorithm (with optimal parameter setting)

### Table 24: Top 10 results k-NN algorithm

**k-NN (k=20)**

<table>
<thead>
<tr>
<th>Outlier top 10</th>
<th>Cost</th>
<th>Value</th>
<th>Profit</th>
<th>Profit %</th>
<th>Delivery surplus</th>
<th>Entry user</th>
<th>Entry time</th>
<th>Document date-Entry date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 25: Top 10 results one-class SVM

**One class SVM (nu=0.2)**

<table>
<thead>
<tr>
<th>Outlier top 10</th>
<th>Cost</th>
<th>Value</th>
<th>Profit</th>
<th>Profit %</th>
<th>Delivery surplus</th>
<th>Entry user</th>
<th>Entry time</th>
<th>Document date-Entry date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 26: Top 10 results uCBLOF

**uCBLOF (k=40)**

<table>
<thead>
<tr>
<th>Outlier top 10</th>
<th>Cost</th>
<th>Value</th>
<th>Profit</th>
<th>Profit %</th>
<th>Delivery surplus</th>
<th>Entry user</th>
<th>Entry time</th>
<th>Document date-Entry date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Appendix J: Results two dimensional anomaly detection algorithm with synthetic data

Table 27: Output k-NN algorithm (late invoice payment)

<table>
<thead>
<tr>
<th>Outlier score</th>
<th>Day payment outstanding</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.845</td>
<td>87</td>
<td>Company D</td>
</tr>
<tr>
<td>0.48</td>
<td>60</td>
<td>Company A</td>
</tr>
<tr>
<td>0.43</td>
<td>57</td>
<td>Company D</td>
</tr>
<tr>
<td>0.03</td>
<td>25</td>
<td>Company D</td>
</tr>
</tbody>
</table>

Figure 37: Visualization k-NN algorithm (late invoice payments)

Table 28: Output one-class SVM algorithm (late invoice payments)

<table>
<thead>
<tr>
<th>Outlier score</th>
<th>Day payment outstanding</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.13</td>
<td>87</td>
<td>Company D</td>
</tr>
<tr>
<td>38.08</td>
<td>60</td>
<td>Company A</td>
</tr>
<tr>
<td>37.89</td>
<td>57</td>
<td>Company D</td>
</tr>
<tr>
<td>2.68</td>
<td>25</td>
<td>Company D</td>
</tr>
</tbody>
</table>

Figure 38: Visualization one-class SVM algorithm (late invoice payments)

Table 29: Output uCBLOF algorithm (late invoice payments)

<table>
<thead>
<tr>
<th>Outlier score</th>
<th>Day payment outstanding</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.89</td>
<td>87</td>
<td>Company D</td>
</tr>
<tr>
<td>0.54</td>
<td>60</td>
<td>Company A</td>
</tr>
<tr>
<td>0.431</td>
<td>57</td>
<td>Company D</td>
</tr>
<tr>
<td>0.04</td>
<td>25</td>
<td>Company D</td>
</tr>
</tbody>
</table>

Figure 39: Visualization uCBLOF algorithm (late invoice payments)
Appendix K: Different anomaly detection results with original date (i.e. not normalized)

In Figure 39, 40, 41 are the results of the three different anomaly detection algorithms shown for the theoretical problem of chapter 6. The color of the data points show the outlier score. There can be concluded that the anomaly scores do not provide the expected results. This is caused by the date that has not a normalized value. Therefore data transformation is needed to be able to include the date variable with other variables in one anomaly detection model.

Figure 40: Output k-NN algorithm (date not normalized)

Figure 41: Output uCBLOF algorithm (date not normalized)

Figure 42: Output one-class SVM (date not normalized)
Appendix L: Visualization anomaly detection output (three dimensional problem with synthetic data, payment error)

One-class SVM

Figure 43: Visualization k-NN algorithm (switched account numbers)

Figure 44: Visualization one-class SVM algorithm (switched account numbers)

Figure 45: Visualization uCBLOF (switched account numbers)
Appendix M: Typical debit and credit accounts

A specific account should often contain only debit or only credit postings. For example sales revenues, service revenues and interest revenues are accounts which are normally increased with a credit entry. Sales return and sales discount accounts have the opposite, only debit entries. The same concepts holds for many more accounts. An overview is given in Figure 46.

<table>
<thead>
<tr>
<th>Account Classification</th>
<th>Normal Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assets</td>
<td>Debit</td>
</tr>
<tr>
<td>Contra asset</td>
<td>Credit</td>
</tr>
<tr>
<td>Liability</td>
<td>Credit</td>
</tr>
<tr>
<td>Contra liability</td>
<td>Debit</td>
</tr>
<tr>
<td>Owner's Equity</td>
<td>Credit</td>
</tr>
<tr>
<td>Stockholders' Equity</td>
<td>Credit</td>
</tr>
<tr>
<td>Owner's Drawing or Dividends Account</td>
<td>Debit</td>
</tr>
<tr>
<td>Revenues (or income)</td>
<td>Credit</td>
</tr>
<tr>
<td>Expenses</td>
<td>Debit</td>
</tr>
<tr>
<td>Gains</td>
<td>Credit</td>
</tr>
<tr>
<td>Losses</td>
<td>Debit</td>
</tr>
</tbody>
</table>

*Figure 46: Usual type of postings to account*
Appendix N: Summary interview for the identification of anomaly detection applications.

Identification of anomaly detection applications
Unstructured interview to identify an interesting area to apply anomaly detection on transactional level data.

Goal:
Anomaly detection algorithms are applied to synthetic data to detect exceptional transactions. This application was successful and now is investigated how the same technique could add value in a practical application. The practical application should be in line with the previous analyzes of this study. So, an application on the detection of anomalies in transaction level data will be identified.

Approach:
The knowledge of (IT) auditors about what unsupervised machine learning and anomaly detection can do is very limited. The application of anomaly detection is explained by making use of the two analysis on synthetic data. By showing these problems and results, the interviewees get an idea about what is possible with anomaly detection. By understanding the technique, they are able to link the concept to their domain of expertise; the audit field. In this way, the link is made between the anomaly detection domain and the audit domain.

Interviewees:
Name: Roger Haenen
Function: Manager ITA
Date: 09-09-2016 & 20-09-2016

Conclusions:
The anomaly detection technique is interesting and could help in diverse applications. One clear application is further discussed; Finding ‘Tax anomalies’. Data for this application is available and therefore possible to investigate in this study. Vendors have often hundreds of the same kind of purchase orders of which some surprisingly have a deviating tax code. These transactions are interesting to investigate because in the past these deviations turned out to be errors. It would be nice if these deviating tax codes could be found with a more flexible technique than currently used.
Appendix O: Review output tax anomaly detection problem

Review output tax anomalies
Unstructured interview to review the tax anomalies that result from the anomaly detection model.

Goal:
To evaluate whether the tax anomalies that are found by the model are also interesting in real life, the output will be shown and discussed with an IT auditor from the company this study is conducted in. In this way the meaning of the anomalies in real life is elaborated.

Approach:
The anomaly detection model is shown and explained to the auditor. The data is loaded into Rapidminer and the model is run. The results are shown and discussed.

Interviewee:
Name: Roger Haenen
Function: Manager ITA
Date: 22-09-2016 and 10-10-2016

Conclusions:
“De huidige procedure voor het vinden van tax code anomalies vindt plaats door middel van targeted analytics. Hierbij word de combinatie tax code – factuurnummer per leverancier gecategoriseerd. In de standaard setup wordt er gezocht naar facturen met tax codes die minder dan 5% voorkomen, en voor 95% een andere tax code wordt gebruikt. Hierbij wordt de aannemen gedaan dat in deze 5%-bucket een verkeerde tax code wordt gebruikt. Nadeel van deze analyse is dat bij een beperkte set aan facturen mogelijke anomalies niet worden gedetecteerd. Dit kan worden opgelost door het 5%-criterium op te rekken naar 10% of 15%, met als beperking dat hierbij het aantal ‘false positives’ toeneemt.

Ja, toegevoegde waarde. De tax anomalie analyse leent zich bij uitstek voor een detectie algoritme i.p.v. een ‘targeted’ test. De feedback van de end-user om aan te geven of een anomalie terecht door het algoritme wordt opgepakt, is mijns inziens van toegevoegde waarde.

Een verkeerd gekozen tax code leidt tot een onjuiste btw-aangifte – waarbij het zowel een btw-risico (onterecht geclaimde input btw-) als een btw-kans (gemiste terug te claimen btw) kan betreffen.

Bruikbaar, mits een zorgvuldige review door de eindgebruiker. De daadwerkelijk opvolging of een anomalie terecht door het algoritme is aangemerkt, is cruciaal. Indien ja, dan kwantificeert het algoritme de btw-kans en the btw-risico bij het onjuist selecteren van de tax code.

Mogelijkheid tot feedback van de eindgebruiker om het algoritme bij ‘te sturen’ om direct tot een scherpere kwantificatie van de btw-risico’s en kansen te komen zou van toegevoegde waarde zijn.”

74
### Appendix P: Validation k-NN anomaly detection algorithm – assessment of interestingness of top 30 transactions in output anomaly detection model

**Goal:** Investigate what the output of the anomaly detection algorithm means for auditors in a more specific assessment of the individual transactions.

**Interviewee:**
- **Name:** Margriet van Eenbergen
- **Function:** Management Trainee audit
- **Date:** 02-11-2016

**Table 30: Validation point anomaly detection model output**

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>Interesting transaction: Check transaction details/documents (with client)</th>
<th>Interesting transaction: Check underlying process</th>
<th>Interesting transaction: Provides insights but probably no action required</th>
<th>Non interesting transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>157267-10</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157267-20</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157204-10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160252-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160802-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159043-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159042-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160868-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158050-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158100-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159678-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158916-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160833-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157842-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159677-20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160775-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158166-20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158166-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160942-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160943-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>156662-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157104-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157104-40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157104-20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157104-30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>
### Table 31: Validation point anomaly detection model output (2)

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>Interesting transaction: Check transaction details/documents (with client)</th>
<th>Interesting transaction: Check underlying process</th>
<th>Interesting transaction: Provides insights but probably no action required</th>
<th>Non interesting transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>157267-10</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157267-20</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157204-10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160252-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160802-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159043-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159042-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160868-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158050-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158100-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160868-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159042-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159042-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158916-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159678-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160833-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157842-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159677-20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160775-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157842-20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157842-30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157677-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>159783-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158166-20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>158166-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160942-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>160943-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>156662-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157104-10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157104-40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157104-20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>157104-30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>