Argument Based Machine Learning in an Auditing Setting

Rosa Sterkenburg - 5576296

First supervisor: dr. Floris Bex
Second supervisor: dr. Matthieu Brinkhuis
Daily supervisor: ir. Arthur Verkerke

February 25, 2019

Contents

List of Figures

List of Tables

1 Introduction

Machine learning has great potential for application in the financial domain. The International Auditing and Assurance Standards Board is exploring the possibilities of the use of data analysis in audit [13]. Furthermore, in recent years many papers have been published about using data analysis for fraud detection [20] or for anomaly detection in the financial domain [1].

While the possibilities of machine learning techniques are being recognized, accountants have their reservations. Before they embrace new techniques, it is very important to them that they can understand what is happening so that they can adequately judge the risks of the new techniques themselves. Accountants feel they cannot say that they "are reasonably sure there are no problems with the financial statements" if the only evidence they have is the outcome of an algorithm they do not understand. Virdhagriswaran [27] notices that:"In the financial accounting world, the accountants [...] do not accept 'black-box' approaches."

One possible and promising way to make sure accountants feel comfortable enough to use machine learning is to include argumentation of domain experts during the creation of a model [6, 12]. Argumentation can be added to the learning algorithm in such a way that domain knowledge can be easily incorporated. The benefit of the argumentation could be that the algorithm performs better, but it could also make the results easier to understand and interpret. This thesis will try to apply the ideas of Argument Based Machine Learning to the field of auditing and accounting.

Applying the existing Argument Based Machine Learning techniques to accounting is not straightforward. The literature on Argument Based Machine Learning focuses mainly on supervised rule-learning algorithms [17, 18], while applications in accounting use mostly unsupervised clustering techniques [1, 20]. This thesis aims to partly fill this gap in the literature, by combining clustering and argument based machine learning.

1.1 Problem case

Auditing is the systematic, independent reviewing of the books and accounts of an organization in order to ascertain to what extent the financial statements present a true and fair view of the concern. This means the auditor reviews the financial processes and considers what checks are built in to these processes to prevent fraud and mistakes. An example of such a check is the three-way match, that ensures that a bill can only be paid if there is also a purchase order and a proof of delivery. Finally, the auditor reviews whether the checks mitigate all the known risks, and whether the checks have the desired effect.

This research will focus on one particular part that is important during the audit process, namely that of general ledger entry testing. Right now a considerable amount of time goes to the manual testing of general ledger journal entries [25]. Every journal entry needs to be booked to a specific general ledger account, and there are manual tests done to verify this is done in a correct way. Clustering can be used as a tool to group similar journal entries together, which could be used to identify to which general ledger account the journal entries should probably belong [4].

Data analysis on general ledger entries can be beneficial in two different ways. It can provide accountants with more insight in the data, laying bare patterns that were previously unseen. This extra knowledge helps the accountant to better understand the data, which can lead to a better risk assessment and to a more efficient audit. Furthermore, data analysis can directly be used to detect anomalies, which can represent mistakes or even fraud.

In this thesis argument based machine learning will be applied in an audit setting. A problem and data set have been provided by the audit service of the Dutch central government. The problem in case is to find the general ledger entries that belong to the hiring of external personnel.

Before we go into the specifics of the problem, it is good to outline the context in which the problem arises. In recent years the Dutch government has made considerable cuts to the budget available for personnel, while the amount of government tasks did not lessen. As an effect the same amount of work now has to be done with less people. Since this is not always possible, the possibility exists that more external personnel is hired to get the work done. To know whether this actually happens, a new requirement is added to the annual reports. Each ministry has to provide an appendix to the annual reports that specifies how much is spent on the hiring of external personnel. On the scale of the whole budget of a ministry, the cost of external hiring are negligible, and therefore external hiring did not receive much attention from auditors in the past. Now, however, the auditors need to vouch for the truthfulness of the appendix, and therefore they need to validate the correctness of the general ledger entries concerning external hiring. Unfortunately, it is known that the entering of external hiring in the accounting system is prone to mistakes. The reason for this is that there is a subtle but important distinction between the external hiring of personnel on the one hand, and the outsourcing of specific tasks on the other. Hiring a person to do a specific job, like cleaning the offices or writing a research report, is outsourcing. Hiring a person to be part of the team and do all sorts of things is the hiring of external personnel. Furthermore, there might be reasons to intentionally enter external hiring wrongfully into the books, for example because management does not want to be seen as hiring too many people. This makes verifying the truthfulness of the appendix a difficult task.

To aid auditors in their assessment of the external hiring appendix, we aim to find general ledger entries that describe external hiring, but are not booked as such, or entries that are booked as external hiring but are actually about something else. In this thesis, argument based machine learning will be applied to find general ledger entries about external hiring that probably contain mistakes.

1.2 Research question

As was explained above, the gap in the literature is that argument based machine learning is not applied to clustering. However, applications in audit often use clustering and could possibly benefit from argumentation. This leads to the following research question:

Research Question: Can argument-based machine learning and clustering be combined so that they complement each other and thereby improve the use of clustering in an audit setting? To answer this research question the following sub-questions need to be addressed:

1. What would constitute improvement of the use of clustering for the auditors themselves?
2. What domain arguments and counterarguments do the auditors think should be included in a model to find external hiring entries that might contain mistakes?
3. How do the results of a clustering model change when it is combined with argumentation?
4. Does a model that uses argumentation meet the improvements the auditors formulated?

In chapter 2 the existing literature on clustering in audit and argument based machine learning is reviewed. Chapter 3 details the methods that were used. Each section in chapter 3 deals with one of the above sub-questions. In chapter 4 the obtained results are presented, again with one section for each sub-question. Chapter 5 provides a discussion of the methods and the results, and chapter 6 concludes.

2 Theory and literature

In this chapter the theoretical background of clustering is discussed. Furthermore, an overview of what has already been done with clustering in audit and argument based machine learning is given. The chapter starts with a short summary of the two most used clustering techniques, K-means clustering and hierarchical clustering. In the next section several studies about clustering in audit are highlighted. In the final section the current research on argument based machine learning is summarized.

2.1 Different clustering methods

Clustering is the dividing of a data set into multiple groups, called clusters, such that the members of a group are more similar to one-another than they are to members of the other groups. Clustering is an unsupervised technique, meaning that it is not necessary to provide a labeled data set. Clustering is often used since in most real life problems a labeled data set is not available. This is also the case for this research. While the journal entries are labeled as 'external hire' or 'other', this is not the label we need. What we actually want are the labels 'correct' and 'mistaken', and those are not available.

Clustering is used in the financial domain mostly to identify anomalies; data entries that are dissimilar from the rest of the data set. These anomalies could represent very different things depending on the context, and can range from mistakes in the general ledgers to possible fraudulent credit card payments, to transactions involving money laundering activities. All these things have in common that we expect them to be less frequent than their genuine variants and that we expect them to be noticeably different from normal instances. This means that if we group similar items together, we expect the mistakes or the fraudulent cases to be in small groups of their own, to not be part of any cluster or to be a poor fit in the cluster they are assigned to [22]. Because of this, clustering can be employed to find these mistaken or fraudulent instances.

Chandola et al. [5] define three types of anomaly:

1. Point anomalies
2. Contextual anomalies
3. Collective anomalies

A point anomaly is an individual instance that is remarkably different from the rest of the data. In the set of your personal expenses, a point anomaly could be an unusually high payment. A contextual anomaly is an instance that by itself is not noteworthy, but it is noteworthy in its particular context. An example could be a purchase at your local supermarket. In normal circumstances, this is not anomalous, but it is anomalous if you are on vacation and all your other purchases made in that period are in Spain. Finally, a collective anomaly is a small subgroup of data, with instances that are similar to each other but very different from the rest of data. A bunch of transactions of €0,01 each would be a collective anomaly.

Which kind of anomalies are found with clustering depends on the clustering technique that is used. Usually, point anomalies are defined as instances that do not fit any cluster, while collective anomalies are identified with very small clusters.

K-means clustering is an iterative method that minimizes within cluster distance for a given distance measure. The first step of the K-means algorithm is to randomly assign k cluster centres and to assign each data point to its nearest cluster centre in order to form an initial cluster assignment. The next step is to recalculate the cluster centres as the mean of all the data points assigned to that cluster, and to update the cluster assignment based on the new cluster centres. This is iterated until the cluster assignment no longer changes. The k-means algorithm converges to a local optimum. It is often repeated multiple times to find a (more) global optimum.

A big advantage of k-means clustering is its linear complexity, which makes it applicable to very large data sets. Furthermore, k-means clustering assigns every data point to a cluster, and tends to create clusters of similar size. This can be positive or negative according to the intended use. Downsides of k-means are that it can only deal well with numerical data, and that the number of clusters needs to be known in advance.

Hierarchical clustering is a method where a hierarchy of nested clusters is created. This can be done in two ways. The first is divisive hierarchical clustering, where the data set is split recursively into smaller clusters, until every cluster consists of only one instance. The second method is agglomerative hierarchical clustering, where each observation starts in its own cluster, and clusters are merged when moving up the hierarchy. The output of hierarchical clustering is a dendogram, which is a tree-like diagram that shows how the clusters are nested. The dendogram can be cut of at any point to create the desired amount of clusters.

Advantages of hierarchical clustering are that the number of clusters does not need to be known in advance, but can be determined after creating the dendogram. Furthermore, it is easy to interpret and use, and unlike K-means it finds a global solution, so for the same input it will always provide the same output. The main disadvantage of hierarchical clustering is its complexity. Run time is generally of $O(n^3)$ and it requires $O(n^2)$ of memory, which makes it slow even for medium data sets, and completely unfeasible for bigger data sets.

2.2 Clustering in Audit

This section summarizes some papers from the field of clustering in audit. These papers were selected from a survey of clustering based financial fraud detection [20] and a survey of anomaly detection techniques in the financial domain [1].

Vasarhelyi et al. [26] apply K-means to a refund transaction data set in order to identify anomalies that indicate fraudulent refunds. Three forms of anomalies are identified. Anomalies can be instances that do not belong to a cluster, while all normal instances do form clusters. Anomalies can be instances that are far from the center of their closest cluster, or anomalies could form small clusters, while the normal instances are in big clusters. Not every clustering technique can find all forms of anomalies. Since K-means tends to produce clusters of similar size and assigns every

instance to a cluster, the second option for defining an anomaly, as an instance that is far from its closest center, is the best fit if one wants to use K-means clustering. Vasarhelyi et al. note that this is also the most widely used in fraud detection.

Thiprungsri et al. [22] apply K-means to life insurance claims in order to detect outliers that might be interesting to auditors. After consulting with domain experts, only two features are used for clustering. This is the ratio of interest payment to beneficiary payment and the average number of days between death and the payment date. Both these features were not directly present in the data set but were created using the original data. K-means is chosen because all features are numerical and K-means is simple and deals particularly well with numerical data. Anomalies are identified in two ways. First, small clusters that contain less than 1% of the data are identified as collective anomalies. Second, instances that are far away from their respective cluster centre are identified as point anomalies.

Jans et al. [15] use K-means clustering to find anomalies related to two different types of fraud. They search for fraudulent double payment of invoices, and for fraudulent changes in purchase orders after they are released. It is mentioned that the clustering does not find fraud, but merely clusters that are interesting for further inspection because they differ from the rest and therefore are likely to contain mistakes or cases of fraud. The features that are used for clustering do not come directly from the accounting system, but are ratios derived from the available data. The reason for this is that a high number of double invoices for a user is not an indicator for fraud on itself, but a high number of doubles compared to the total number of invoices for this user, is. In all their experiments, at least one cluster is formed that is significantly smaller than the other clusters. These small clusters are considered anomalous and are selected for further inspection.

An interesting example of a study where hierarchical clustering is used is carried out by Torgo et al. [21, 23]. They use hierarchical clustering for anomaly detection in order to provide auditors with a list of cases that may need further inspection. Furthermore, they want to rank the anomalies based on how likely it is that he case is fraudulent, the cost of further inspection and the benefits when it is found that the case is fraudulent. In case of limited inspection resources, such a ranking can help choose which cases should be inspected. To determine whether an instance is an anomaly, agglomerative hierarchical clustering was applied. This was chosen because, presumably, an anomaly offers more resistance to being merged, and this should be reflected in the merging process. A case study is carried out on foreign trade transactions, and the feature that is used for clustering is a cost/weight ratio.

Another interesting study is the research by Ghani et al. [10]. Ghani et al. have developed a system that detects mistakes in the processing of health insurance claims. A classifier returns claims that probably contain an error. These mistakes need to be corrected, which often takes a lot of time because the auditor needs to find out why a case is flagged as a possible mistake. Clustering is proposed to find clusters of claims that are erroneous for similar reasons. While it is not specified what kind of clustering algorithm is used, the idea of clustering possible mistakes to make the auditing process more efficient is quite interesting.

2.3 Argument based machine learning

This section reviews multiple argument based approaches to machine learning, taking a survey on argument based machine learning [6] as the starting point. However, before argument based machine learning is discussed, some basic concepts from argumentation theory are explained.

The basic notion of an argument dates back to ancient Greece, with Aristotle's syllogisms [3]. An argument consists of premises and a conclusion, where one or more of the premises could be conditional statements. Throughout history, the focus has long been on deductive arguments, leading to standard deductive logic [16]. In a valid deductive argument, the truth of the premises guarantees the truth of the conclusion; if the premises are true, then it is not possible for the conclusion to be false.

However, in practical applications it is not always the case that the truth of the premises guarantees the conclusion. The famous is example is that birds can fly and Tweety is a bird, and therefore Tweety can fly. The conclusion that Tweety can fly is rationally compelling, but for example not true in the case that Tweety is a penguin, since penguins cannot fly. Defeasible reasoning deals with arguments that are rationally compelling but not deductively valid, for example because exceptions to the general rule are possible. Furthermore, it allows for non-monotonicity, meaning that new information can prove an old conclusion no longer true. This is a desirable treat for common sense reasoning, since in real life new information, like the fact that Tweety is a penguin, often makes us change our minds and reconsider our conclusions. This can have the effect that old conclusions, such as the fact that Tweety can fly, are no longer true.

Defeasible reasoning can lead to conflicts, as in the Tweety example where two contradictory conclusions could be drawn. We could say Tweety can fly because he is a bird, or we can say that he cannot fly because he is a penguin. In cases like this, a way is needed to resolve the conflict and evaluate which conclusions are warranted with respect to our complete knowledge base. There are two types of semantics often used for this evaluation; argument frameworks with extensions by Dung [7] and dialectical trees [9].

Argument frameworks by Dung [7] consist of a set of arguments and an attack relation between the arguments. The argumentation framework can be represented by a directed graph, where the nodes are the arguments and the edges represent the attack relation. To determine which conclusion is supported by the arguments, semantics are provided in the form of extension sets. A set of arguments S defends an argument A if every argument that attacks A is in its turn attacked by an argument from S. We can define a function $F(S) = \{A \mid S \text{ defends } A\}$. A complete extension is a set of arguments such that $S = F(S)$. A stable extension S is a complete extension that is maximal w.r.t. set-inclusion and that attacks all arguments not in S. An argument is skeptically accepted if it is in all stable extensions. An argument is credulously accepted if it is in some stable extension. An argument is rejected if it is in no stable extension. An example can be found in figure 1

Dialectical trees start with a notion of defeat between arguments. It is based on the idea of a conversation, where each argument is countered by another argument. A tree is built with at its root node the possible conclusion. Each child node in the tree is an argument that defeats its parent node. A node in the tree is undefeated if it has no child nodes that are undefeated. A node is

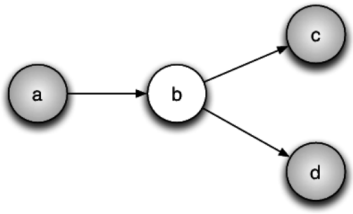

Figure 1: Example of an argument framework. The set {a,c,d} is a stable extension

defeated when at least one of its children is undefeated. The conclusion from the tree is accepted when its root node is ultimately undefeated. See figure 2 for an example of a dialectical tree.

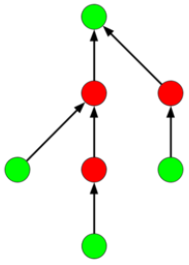

Figure 2: Dialectical tree

As can be seen from figures 1 and 2, extension set semantics and dialectical tree semantics show quite some similarities. Both use graphs where the nodes represent arguments and the arrows depict the attack relation between arguments. Differences are that the graphs that depict the argument frameworks for extension sets do not have to be trees, and that in an argument framework each node represents an unique argument, while in dialectical trees the same argument can be used in different branches.

ABML by Možina et al.
Argument Based Machine Learning as introduced by Možina et al. [17] focuses on rule-based classifiers. A rule-based classifier is a classifier that learns if-then statements from the data and uses those to classify data-entries [24]. Možina et al. formalize arguments as if-then statements, so that they can fit easily within an existing rule-based algorithm.

The main idea is to first train a classifier. With k-fold cross validation that is repeated n times the entries that are most often misclassified are identified. These entries are then given to an expert, who gives an argument why a specific entry belongs in its category. Arguments are statements like "This entry belongs in category 1 because of features A and B", or "This entry belongs in category 2 despite feature C". The next step is to retrain the classifier, taking the constraints provided by these arguments into account.

According to Mozina et al., adding arguments has two benefits. First, it should improve the

classification, resulting in less mistakes. Furthermore, it should result in a more intuitive model, since domain knowledge is directly incorporated in to the algorithm.

The benefit of using such a simple formalization of an argument is that it can easily be incorporated into the algorithm. It also allows for argumentation on the level of a single entry, creating the opportunity of using domain knowledge without having to formulate global domain truths.

The downside of such a simple formalization of argumentation is that it does not offer a way to deal with conflicting rules. Intuitively, when two rules give contrary recommendations an argument should be able to provide a reason to prefer one category over the other. ABML by Mozina et al. however does not use argumentation to resolve such a conflict

Agents that argue and explain classifications (CLA)
Amgoud et al. [2] also combine argumentation and classification. They note that one of the advantages of the combination is that it can not only provide classification, but can also give the reasons behind the classification in a way that is easy to grasp. Another advantage is that an argumentation based approach can handle inconsistent training examples.

Amgoud et al. identify two forms of arguments. The first sort of argument in favour of classifying x in category c is that there exists a training example that classifies x as c. The second sort of argument in favour of classifying x in category c is that there exists a hypothesis h that classifies x in c.

Sentiment polarity classification using argumentation (CleAr)
Carstens et al. [4] use argumentation in the domain of sentiment polarity classification. The task they face is to determine from a given snippet of text whether it is positive or negative. The main advantage of using argumentation is that it makes the classifier applicable to a broader domain. Normally, a sentiment polarity classifier only performs well if the texts it needs to classify are from a very similar domain as the texts it has been trained on. Adding domain-independent argumentation vastly improves out-of-domain classification.

Arguments are words together with a polarity and a score. An example of an argument can be (worthless, negative, 0.4). A total of 6815 arguments are used. The semantics are not one of the two options explained above. They use quantitative semantics, assessing the strength of an argument numerically. Arguments start with a base score, a number between zero and one. This score is then adjusted based on the scores of attacking and supporting arguments.

The argumentation is integrated in the classification process by first letting the classifier provide a label, and then use the formal argumentation to argue about that label. The base score of a label can usually be extracted easily from the classification. If after argumentation the score of the label given by the classifier is lower than the score of the opposite label, the label is changed. This can be interpreted as that when we have sufficient reason to doubt the classifier, we change the label.

Integrating defeasible argumentation with fuzzy ART neural networks (A-ART)
Gómez et al. [11] use argumentation for classification with fuzzy adaptive resonance theory. Fuzzy ART creates overlapping clusters, that are labeled as either negative or positive. A new instance is classified according to the label of the cluster it belongs to. If a new instance belongs to multiple

clusters that disagree about the label, it is hard to classify. Usually such a case is solved by making a random choice. Gómez et al. propose to use argumentation as a means to make such a choice between overlapping clusters.

An argument as used by Gómez et al. is a set of logical formulas such that there exist a defeasible derivation from the argument to the conclusion. A defeasible first-order logic is used to describe arguments like for example: "smaller, more specific, clusters should be preferred over bigger clusters". A counterargument to an argument A is an argument that has as conclusion the negation of one of the formulas in A. For semantics dialectical trees are used.

Gómez et al. note that classification relies mostly on the quantitative aspects of the data, while argumentation uses non-numerical, qualitative information. The combination allows for taking both forms of information into account.

Multi agent inductive concept learning (MAICL)
Ontanon et al. [19] show that the task of inductive concept learning can be expressed as a defeasible first-order logic. Since argumentation can also be expressed in defeasible logic, the logical model of induction allows for a natural combination of the two. Ontanon et al. apply this combination of inductive learning and argumentation to a case of multi-agent ICL. For semantics they use dialectical trees.

Argument accelerated reinforcement learning (AARL)
Gao et al. [8] use argumentation to enhance reinforcement learning in the context of RoboCup Soccer. Agents argue about the best action to undertake in the game .

Arguments are if-then statements of the form "IF premise(A) THEN conclusion(A)". The conclusion of an argument A is the recommended action, while its premise describes under which circumstances the argument is applicable. An example of an argument can be "If agent T is the one closest to the ball, than T should try to tackle and take the ball". Semantics are provided by extensions.

Gao et al. specifically note that domain knowledge can be error-prone or even self-conflicting, and that argumentation is a good way to deal with conflicting domain knowledge.

Summary of argument based machine learning approaches
As can be seen from the above summaries, most approaches on argument based machine learning use a notion of conflicting arguments with an attack or defeat relation between arguments. In combination with semantics like the extension sets or dialectical trees, this allows one to draw conclusions in case of conflicting arguments. Mozina et al. [17] are the only exception by handling just arguments but no relation between the arguments. Two of the papers [2, 17] mention explicitly that argumentation should be able to enhance the understanding of users of the machine learning application. Argumentation can thus help to tackle the "black box". However, none of the discussed papers deal with unsupervised learning. Only Gomez et al. [11] use a clustering technique, but they combine this with class labels to arrive at a technique that is semi-supervised.

A summary of the argument based approaches can be found in table 1.

	Type of ML	Semantics	Type of arguments	Advantages as stated by the authors
[17] ABML	Rule-based classifier	-	if-then statements	arguments do not need to be true for the entire domain, but can apply only in specific cases
[2] CLA	classification by arguments	Extensions (Dung)	examples and hypotheses	
[4] CleAr	Sentiment Polarity Classification	Quantitative	6815 words with labels	domain independence
[11] A-ART	Fuzzy Adaptive Resonance Theory Neural Networks	Dialectical trees	sets of formulas	allows for the use of both qualitative and quantitative information
[19] MAICL	inductive concept learning	Dialectical trees	-	
[8] AARL	Reinforcement learning	Extensions (Dung)	if-then statements	deals with conflicting domain knowledge

Table 1: Summary of argument based machine learning approaches

3 Method

This chapter details the methods that were used to answer the four sub-questions that were formulated in the introduction. In the first section the available data is described, as well as how that data was transformed. The next four section deal with a sub-question each.

3.1 Description of the data and transformation of the data

The data set consisted of all financial transactions of the ministry of foreign affairs in 2017, totaling 2,120,803 records. A filter was applied to select only the rows that deal with expenses. This resulted in a data set with 257,274 records. Each record represents a journal entry in the general ledgers, and from these 257,274 journal entries, 7519 are booked as external hire. The data contained 114 columns. While it is not informative to list all 114 columns, the subjects of these columns can be roughly placed in nine categories:

- Document ID
- Date and time
- User
- Description
- Place in the budgetary structure
- Cost category
- Debtor
- Creditor
- Amount

Columns in document ID contain information that together uniquely identifies the transaction. Date and time specify when the transaction was entered into the accounting system, and when it was actually payed. User specifies the person who entered the transaction into the system, and which department they work at. Description contains various free fields that describe for what purpose the cost is made. For example, the description fields of a transaction to a lawyers firm will tell which lawsuit this transaction concerns. Place in the budgetary structure tells to which chapter, section and subsection the cost was booked.The budget of each ministry has several chapters, corresponding to the different tasks of that ministry, and each chapter is further subdivided into sections and subsections, further specifying the tasks and goals. Place in the budgetary structure thus tells to which tasks or goal the cost is booked. Cost category does not specify for what purpose the costs are made, but what kind of cost it is. Examples of cost categories are personnel, material or housing. Debtor and creditor specify the other party in the transaction. For every transaction, only one of these is filled. Since a filter was applied to select only the rows that deal with expenses, the debtor columns are mostly empty. Finally, the amount columns specify the amount in local currency and in euros, which (if any) foreign currency is used, and what the conversation rate was at the time of the transaction.

Based on what aspects of the data set various experts mentioned as useful for identifying external

hiring journal entries, the following 13 columns were selected to use in analysis:

- Transaction code
- Amount
- User name
- User group
- Booking key
- Cost centre
- Document type
- Operational goal
- Budgetary position
- Creditor
- Description
- Text booking
- Document header

Transaction code is a code that describes the transaction type, for example automated salary payment or manual payment. The amount is the amount in euros. The user name is an a code that specifies which employee entered the transaction into the accounting system. User group specifies which administrative department entered the transaction into the accounting system. Booking key determines whether it is a debit or a credit booking. Cost centre tells which department made the costs. An example of a cost centre could be the embassy in Rome. Document type states what type of document initiates the transaction. This could be something like a bill or a receipt. Operational goal states to which goal from the budgetary structure the transaction is credited. Budgetary position states to which particular subsection of that goal the transaction is credited. Creditor gives the name of the creditor. This can be both a company or an individual. Description is a free field, were any extra information about the transaction can be entered. The same holds for text booking. Document header gives the first row of the document that initiated the transaction.

From this list only transaction code was not mentioned by the interview respondents. It was included because it contains information about the type of transaction, which, according to the data analysis team of the audit department, is relevant to distinguishing external hiring journal entries.

Except for amount, all the chosen variables were categorical variables. That meant they needed to be transformed to a numerical variable to be able to use them for K-means clustering. The conventional option to convert categorical variables, the creating of a binary variable for each category, was unfeasible because of the high amount of distinct categories. This would have resulted in about 22000 dummy variables, mostly due to the high amount of distinct creditors.

It was suggested by the data analysts from the audit department that the categorical variables could be converted by relative frequency. It is the frequency of how often a specific category is used for an external hire journal entry. For example see the frequency table given by table 2. This is the frequency table for document type versus external hiring. For clustering we will use the numbers in the column external hiring. Except for the text fields with descriptions, every variable was transformed in this way.

	Other journal entries	External hiring journal entries
AB-Boekhoudingsdocument	5241	36
DA-Debiteurdocument	4	0
DG-Debiteur creditnota	15	0
DR-Debiteur factuur	1	0
DZ-Debiteur betaling	1	0
GM-GM Crediteur Factuur	8113	0
KA-Crediteurdocument	10426	114
KG-Crediteur creditnota	266	0
KP-Rekening verzorgen	7	5
KR-Crediteur factuur	146289	949
PP-Doorboeking HR-FICO	30242	0
RE-Factuur bruto	93	181
SA-Grootboekrek.doc.	1038	13
SB-GrBkRek.boeking	5296	28
SK-Kasdocument	40011	194
TD-TEM Debet (Factuur)	8	0
VL-TEM VLP bericht	426	0
WE-Goederenontvangst	2278	5999

Table 2: Relative frequencies of Document type

From the three fields that contained descriptions; description, document header and text booking, two new variables were formed: External_hire_in_description and Keyword_in_description. External_hire_in_description is 1 if one of the three text fields contains the terms external hire and 0 otherwise. Keyword_in_description is 1 if the text fields contain any of the terms 'advice', 'week' or 'hour', and 0 otherwise. The keywords were provided by the data analysts from the audit department. After transformation, the final set of features used for analysis was:

- Transaction code frequency
- Amount
- User name frequency
- User group frequency
- General ledger account frequency
- Booking key frequency
- Cost centre frequency
- Document type frequency
- Operational Goal frequency
- Budgetary position frequency
- Creditor frequency
- External_hire_in_description
- Keyword_in_description

3.2 Interviews to determine what auditors see as improvement

The first sub-question is "What would constitute improvement of the use of clustering for the auditors themselves?". To answer this question, semi-structured interviews were conducted. A semi-structured interview is an interview were deviation from the questionnaire is allowed, so that new ideas that arise as a result of what the interviewee says can be taken into account.

Four respondents were consulted. All interviewed experts were financial accountants who work at the ministry of Foreign Affairs. We have chosen to interview only financial auditors from the ministry of Foreign Affairs, since auditors from Foreign Affairs have noted that the mistake of exchanging external hire for outsourcing and vice versa happens quite often and they would benefit much from a tool that helps to find these mistakes.

The interview consisted of two parts. The first part questioned the current use of data-science in the financial control and what risks accountants see to the use of data-science. Based on the risks the accountants identified, they were asked about what they say as possible improvements to minimize those risks.

The second part focused on the problem of finding possible mistakes in external hiring journal entries. Respondents were asked whether they thought data-science could be useful with respect to this problem. Furthermore they were asked how external hiring is currently audited and how they thought data-science could improve this process.

The questionnaire used as the basis of the semi-structured interview can be found in appendix A.1.

3.3 Questionnaires for collecting arguments

The second sub-question is "What domain arguments and counterarguments do the auditors think should be included in a model to find external hiring entries that might contain mistakes?". To answer this question, structured questionnaires were conducted. However, to design the questionnaires, the precise form of an argument needed to be determined first. The notion of counterarguments was based on the way respondents were asked for arguments. This section first describes the notion of argument, than the questionnaire design and finally the notion of counterargument.

What is an argument?
It was chosen to structure the arguments like the arguments used by Mozina et al. [17]. This means an argument is of the form "This item belongs to category X because feature A has value B". The conclusion of the argument is "This argument belongs to category X", while its premise is "feature A has value B". The premise of an argument is true for a journal entry if for that journal entry the feature mentioned in the premise indeed has the value mentioned in the premise. Furthermore, arguments with just a conclusion and no premises are possible.

The main advantage of formulating the arguments in this way is that it is simple, and therefore it was relatively easy for experts to provide arguments. Furthermore, arguments could be provided based on singular instances. This allowed experts to formulate their arguments case by case, and

relieved them of the very difficult task of formulating arguments that are applicable to all possible cases.

Questionnaire design for collecting arguments
For the collecting of the actual arguments, five financial auditors from the ministry of Foreign Affairs were consulted. Respondents were asked to evaluate 20 journal entries, where each respondent got a different set of journal entries. This way a sample of 100 possible mistakes could be evaluated without asking too much of the time of the respondents. All of the entries provided to experts were anomalous with respect to the context of their assigned cluster, and were therefore labeled as possible mistakes.

First the respondents were asked whether the journal entry concerned external hiring or not. They were not told how the entry was booked in the general ledger, and neither were they told the outcome of the clustering algorithm. There was the option to answer that they did not know based on the given information, but respondents were discouraged to use that too often. If the respondents were reasonable sure, even though they could not be a 100% sure, they were asked to provide an answer. The option that states they do not know was not for cases where they could not be sure, but for cases where they could not make any judgment at all.

Next, the respondents were asked to provide each journal entry with argumentation. For each argument they had to choose its conclusion first, thus whether it was an argument in favour of external hire or not. Furthermore, they had to choose one from the 15 features for the premise of their argument. Finally, there was a free field where they could write what about that feature argued for their conclusion.

They were asked to provide at least two arguments per entry, and encouraged to provide at least one argument with a conflicting conclusion, though the latter was not a hard restriction. The reason to ask for at least three arguments was to ensure that less convincing arguments did not get neglected. Furthermore, respondents might have the tendency to only offer arguments in favour of their conclusion, while reasons to doubt their conclusion are equally valuable for identifying possible mistakes. Therefore they were encouraged specifically to provide at least one argument with a conflicting conclusion.

Finally, respondents were asked to order the arguments they had provided from most persuasive to least persuasive. To keep the task manageable they were asked only to order the arguments given per entry. That meant that they only had to compare arguments with respect to a specific journal entry, and they did not have to compare their relative strength in general. An example question is shown in figure 3.

What is a counterargument? As was explained above, experts were asked to provide multiple arguments per journal entry, and to rank the arguments per entry from most persuasive to least persuasive. From this a partial ordering over all the arguments could be derived. If for one journal entry arguments were given for both conclusions, i.e. the respondent states that it could be external hiring because of feature x but it could be something else because of feature y, these arguments were taken to attack each other. This attack relation combined with the partial ordering results in a defeat relation.

Definition 3.1. *Argument A is said to defeat argument B if argument A and B have contrary*

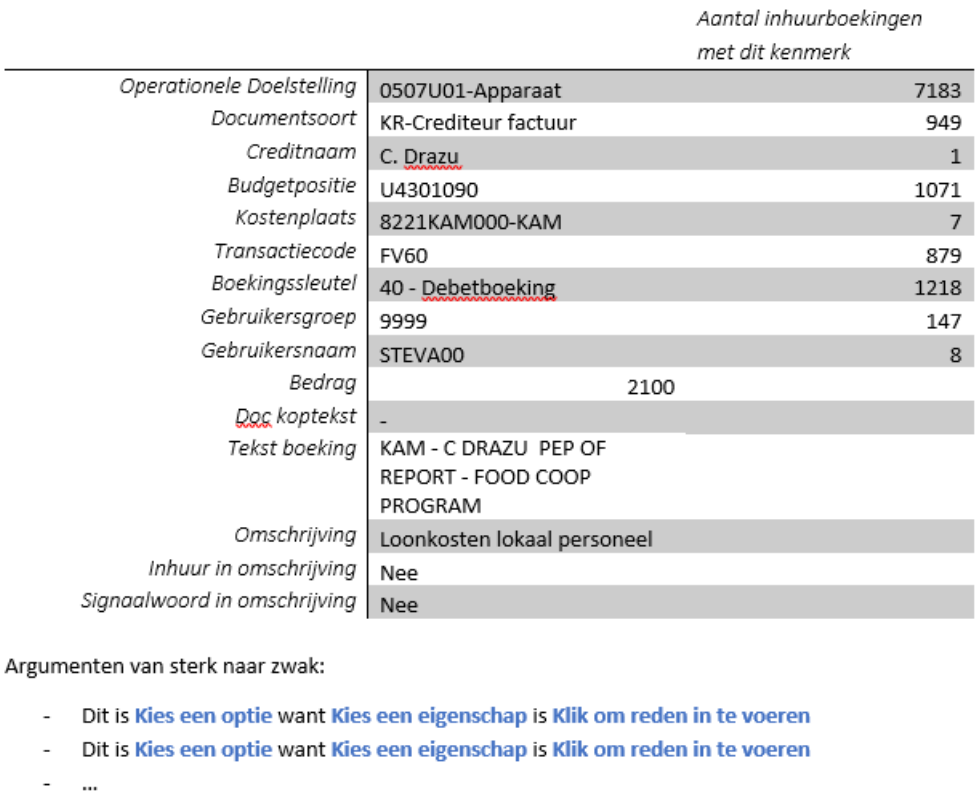

Figure 3: Example of a question in the questionnaire to collect argumentation

conclusions, are given with respect to the same journal entry at least once and argument A is ranked as more persuasive than argument B.

As an example take a journal entry for which two arguments are given. The first argument states that it is not external hire because the amount is below €300. The second states that it is external hire because the description contains a month. The first argument is ranked as more persuasive than the second argument. According to definition 3.1, the first argument is a counterargument to the second argument, but not vice versa.

However, as was noted already in the questionnaire design, respondents might have the tendency to only offer arguments in favour of their conclusion. Furthermore, it was anticipated that they would not often repeat themselves, and therefore very general counterarguments that are often applicable might be missed. Because of that a second notion of defeat was defined:

Definition 3.2. *Argument A is said to defeat argument B if the following conditions are met:*

- *Argument A and B have contrary conclusions.*
- *The premises of arguments A and B are both true for a single journal entry of the questionnaire at least once.*
- *Argument A is mentioned explicitly for a journal entry where the premise of argument B is true, and*
  - *A is ranked as more persuasive then argument B*
    or
  - *argument B it not explicitly mentioned*

For example take two journal entries that have both received one argument in the questionnaire. Journal entry 1 gets the argument (A) that it is external hire because the description contains a month. Journal entry 2 gets the argument (B) that it is not external hire because the amount is below € 300,-. However, for journal entry 1 it is also the case that the amount is below € 300,-, though it is not explicitly mentioned as an argument for journal entry 1. This means that we have two arguments A and B with contrary conclusions and the premises of arguments A and B are both made true by journal entry 1. Furthermore, A is mentioned explicitly for a journal entry were the premise of argument B is true, namely journal entry 1. Therefore, argument A is a counterargument to argument B by definition 3.2.

3.4 Clustering model and argument analysis

This section describes the methods used to answering the third sub-question: "How do the result of a clustering model change when it is combined with argumentation?" To evaluate how the results of a model change when argumentation is added, we need a clustering model, a way to implement the argumentation and a way to combine them.

3.4.1 Clustering

In chapter 2 various ways to use clustering to find anomalies were discussed. The different types of anomalies were point anomalies, contextual anomalies and collective anomalies. Because of the size of the data-set, K-means clustering was the only feasible clustering method. Since K-means assigns every data point to a cluster, we cannot identify anomalies as instances that do not fit any cluster. Furthermore, K-means tends to produce clusters of similar size, so treating small clusters as anomalous is not ideal either.

The idea of a contextual anomaly as mentioned by [22], however, can be used easily in combination with K-means clustering. There were two normal classes, external hiring journal entries and other

journal entries. It was known that both these classes contained mistakes, so while both external hiring and other journal entries could not be considered as anomalous by themselves, a lonely external hiring entry that is grouped into a cluster consisting solely of other journal entries, was considered anomalous because it did not fit the context of its cluster. In clusters that were predominated by other entries, the few external hiring entries were suspect. See figure 4 for a visual representation of this idea.

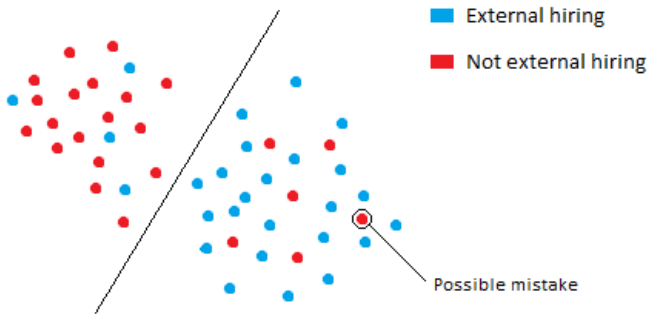

Figure 4: Visual representation of anomalies

The contextual anomalies also gave rise to an interesting quality measure, namely the uniformity of the clusters. If the uniformity of clusters can be increased, meaning that clusters are closer to containing exclusively external hiring journal entries, or exclusively other journal entries, less anomalies will be found. The anomalies that persist in highly uniform clusters, are more likely to indicate actual mistakes because they differ more strongly from their context. See figure 5. This means that the more uniform the produced clusters are, the more useful the clustering is to find mistakes in the general ledger.

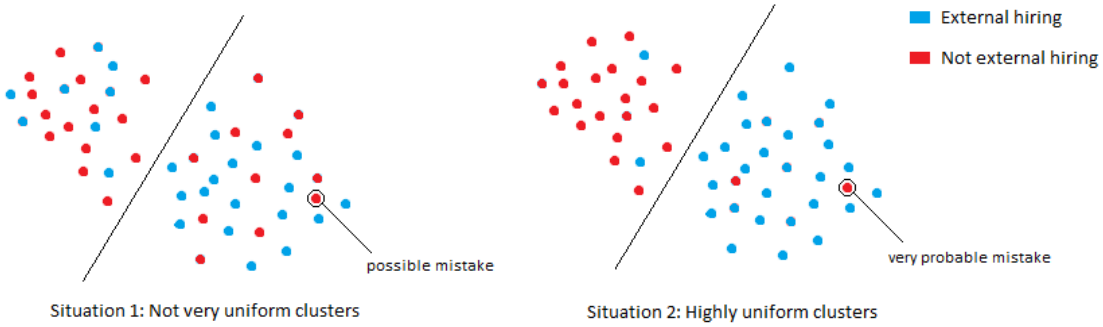

Figure 5: Visual representation of uniformity as quality measure

As was already mentioned above, K-means clustering was the only possible clustering method because of the size of the data-set. Furthermore, because the data-set was so large, even the standard implementation of K-means in R was too slow. Therefore, K-means was implemented using the h2o package in R [14]. h2o is an algorithm optimization package, and considerably reduced run time and memory need.

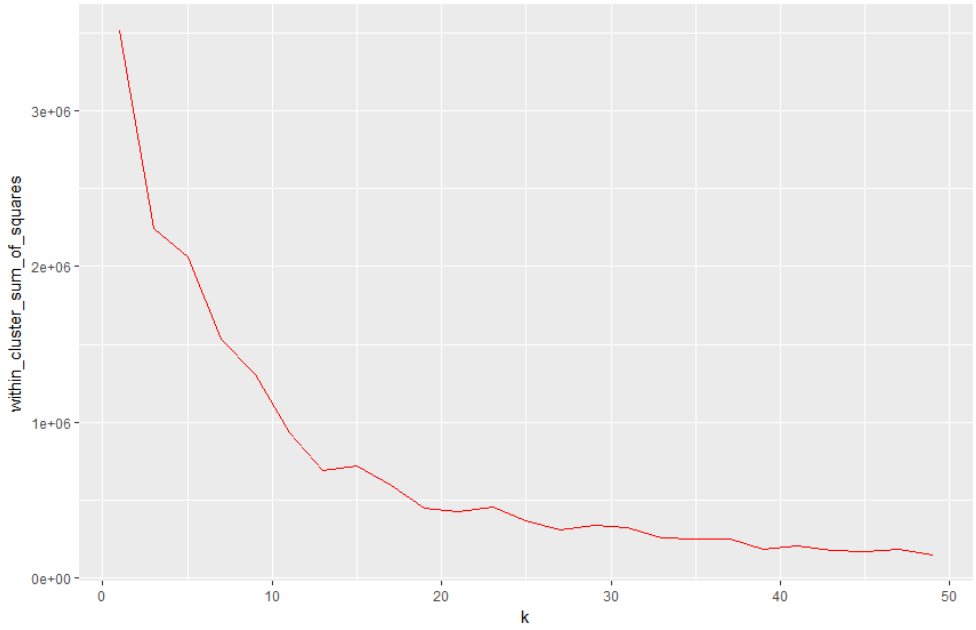

Figure 6: Within sum of squares plotted against the amount of clusters

K-means depends mostly on four parameters:
- K, the number of clusters
- the chosen distance metric
- the way the initial cluster centroids are chosen
- the number of times the algorithm is run before the best cluster assignment is chosen

Figure 6 shows how the often used metric of within cluster sum of squares depends on how many clusters were formed. Figure 7 shows how the uniformity of clusters depends on how many clusters were formed. The green line shows what percentage of the records was placed in a cluster with total uniformity. The red line show what percentage of the records was placed in a cluster with more than 95% uniformity. Based on these two graphs it was chosen to do K-means clustering with 10 clusters.

Table 3 shows the average sum of squares over 20 runs for the different initialization methods. Random initialization randomly chooses k rows of the training data as cluster centres. PlusPlus initialization chooses one initial center at random and weights the random selection of subsequent centers so that points furthest from the first center are more likely to be chosen. Furthest initialization chooses one initial center at random and then chooses the next center to be the point furthest away in terms of Euclidean distance. Based on table 3, PlusPlus was chosen for initialization method.

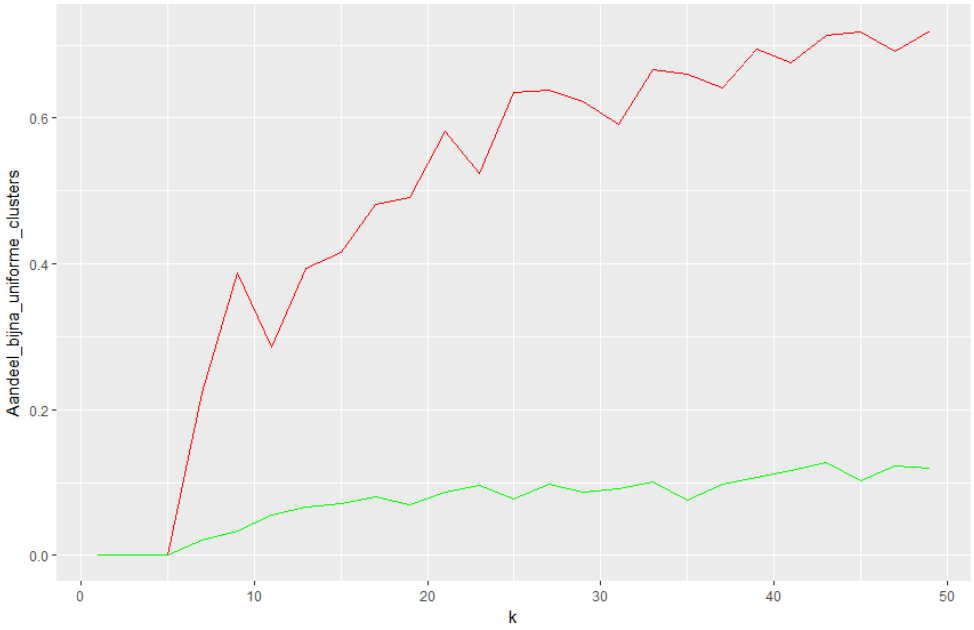

Figure 7: % of uniform clusters plotted against the amount of clusters

	Random	PlusPlus	Furthest
Within cluster sum of squares	1.168.686	1.069.592	1.173.765

Table 3: Different initialization methods

3.4.2 Argumentation

In section 3.3 it was explained that, based on the work of Mozina et al. [17], an argument in this study was chosen to be an if-then statement about a characteristic of one of the features. As was noted by Gao et al. [10], domain knowledge can be self-conflicting. This could result in conflicting arguments, and in those cases a way was needed to determine which conclusions can be drawn from the arguments. Argumentation semantics as explained in section 2.3 provide a way to draw these conclusions, and in this research dialectical trees [11, 19] were used to evaluate which conclusions were warranted. A dialectical tree has an argument as its root node, and the immediate children of every node are counterarguments to the parent node. Nodes are labeled as defeated or undefeated and all leave nodes are undefeated. A node is defeated if it has at least one child that is undefeated, and undefeated otherwise. An argument is accepted if the root node of its associated tree is undefeated.

As the root of a tree, arguments with empty premises were used. These arguments were simply: 'This journal entry is external hire' or 'This journal entry is not external hire'. Every argument with a premise that was true for that journal entry and with a different conclusion than that of the root argument, counted as a counterargument to the root argument. Counterarguments to

arguments that did not have empty premises, were as described by definition 3.2. The conclusion that a journal entry is external hiring was drawn if the argument for external hiring was accepted, while the argument for not external hiring was rejected. Similarly, the conclusion that a journal entry is not external hiring was drawn if the argument with that conclusion was accepted, while its opposite was rejected. An example of two dialectical trees that together warrant the conclusion that this journal entry is external hire can be found in figure 8.

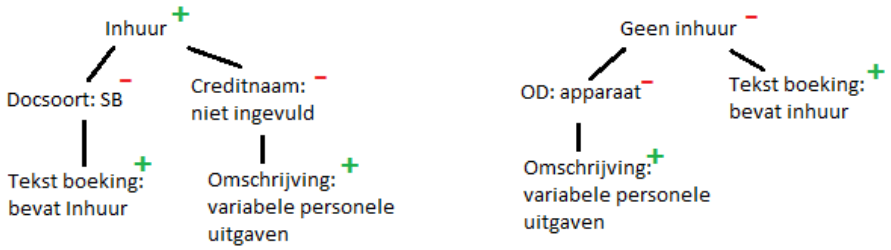

Figure 8: Example of two dialectical trees that together warrant the conclusion external hire

The definition of a dialectical tree calls for evaluation per branch of the tree. However, this is needlessly computationally complex, since if one branch is evaluated with an undefeated node right below the root node, the whole tree will conclude that the conclusion is unwarranted [11]. The only cases where a conclusion is accepted, is if every branch of the three is of odd length, thereby ensuring that all the children of the root node are defeated, and thus that the root node itself is undefeated.

While there exists libraries for the evaluation of a dialectical tree in Java, the same does not hold for R. Therefore an ad-hoc method of evaluating the trees was developed. Based on the observations mentioned above, the method is breadth-first instead of depth-first, meaning it evaluates per layer of the tree. Below the algorithm for the evaluation of the arguments using dialectical trees is given in words. Illustrations of the situation at different steps in the algorithm can be found in figures 9, 10 and 11.

1. Select a journal entry
2. Find all argument that are applicable to this journal entry
3. Start with evaluating the possible conclusion: this is external hiring
   (a) Select all argument from the applicable arguments that have “not external hire” as their conclusion.
   (b) If any of the arguments found in step 3(a) does not have a counterargument in the set of applicable arguments, this branch of the three terminates and the conclusion is “not warranted”.

(c) If all of the arguments found in step 3(a) have at least one counterargument in the set of applicable arguments, select all these counterarguments.

(d) If all of the arguments found in step 3(c) do not have any counterarguments in the set of applicable arguments, all branches of the tree are closed and the conclusion is "warranted"

(e) If any of the arguments found in step 3(c) has at least one counterargument in the set of applicable arguments, select all the counterargument for the arguments from step 3(c) and use these arguments as starting point for step 3(b). Repeat steps (b,c,d) until a conclusion can be drawn.

4. Repeat the same steps for the possible conclusion: this is not external hiring

   (a) Select all argument from the applicable arguments that have "External hire" as their conclusion.

   (b) Follow steps 3(b) trough 3(e)

5. If both conclusions from 3) and 4) are "warranted" or both are "not warranted", no general conclusion can be drawn based on the arguments. If only one conclusion is "warranted", that conclusion is the general conclusion for this journal entry.

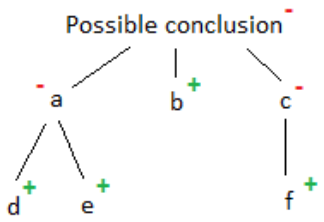

Figure 9: Illustration of the situation at step 3(b). Argument b does not have any counterarguments and is therefore labeled as undefeated. Therefore the conclusion has as least one child that is undefeated, so the conclusion is defeated.

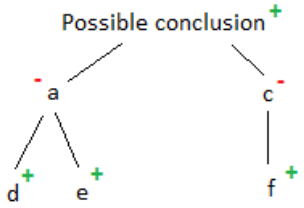

Figure 10: Illustration of the situation at step 3(d). Every child of the conclusion is defeated, since all children have a undefeated counterargument. Therefore, the conclusion is undefeated.

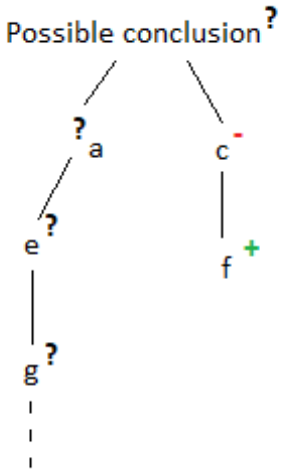

Figure 11: Illustration of the situation at step 3(e). The open branch below argument g makes it impossible to draw a conclusion. Whether the conclusion is defeated or undefeated depends on how argument g is evaluated. Therefore, argument g is treated as the new possible conclusion and the algorithm starts over.

3.4.3 Combining clustering and argumentation

As was mentioned in the introduction, no literature was found about the combination of clustering and argument based machine learning. No method was found to truly integrate clustering and argumentation, and therefore a twofold approach was chosen, inspired by the work of Carstens et al. [4].

Carstens et al. integrate classification and argumentation by using the classifier to provide a label and then argue about that label. The same can be done with clustering, where the label provided by the clustering is the label of the majority of the cluster. Every possible anomalous journal entry per definition will get a different label from the clustering than the label with which it is booked in the general ledger, and we can use argumentation to argue about both these labels. The argumentation could confirm the clustering label, and thereby confirm the fact that the entry is anomalous and in need of extra attention from an auditor. The argumentation could also agree with the general ledger label, and thereby suggest that the flagging of the entry as anomalous is a false positive and that the entry is likely to be correctly booked in the general ledger.

This idea of combining the argumentation and clustering also resonates with the ideas of Torgo et al. [21, 23], who try to rank the anomalies that are found with the clustering from more likely to be actually erroneous to less likely to be actually erroneous in order to make the auditing process more efficient. Arguing about the clustering labels can provide a coarse ranking, since anomalous journal entries where the cluster label is confirmed by the argumentation are more likely to be actually erroneous than journal entries where the clustering label is contradicted by the argumentation. Furthermore, Ghani et. al. [10] aim to use clustering to group similar sorts of anomalies together to make the auditing process more efficient. They want to present auditors with a list of anomalies that are identified as anomalous for similar reasons. This can easily be achieved by argumentation as well, by grouping the anomalies based on which arguments are applicable to the anomalous journal entries.

To summarize, the way clustering and argumentation were combined in this research is a twofold approach. First, clustering is used as a classifier to find anomalies. The next step is to use argumentation as a classifier, to reason about the anomalies, and to use the argumentation to provide extra information by which the anomalies can be judged.

3.5 Expert evaluation

The final sub-question is: “Does a model that uses argumentation meet the improvements the auditors formulated?”. To answer this sub-question, the results of the argumentation on the anomalous journal entries were evaluated by a financial auditor. Unfortunately, due to restraints on the time the auditors were available for this research, it was impossible to let the auditors evaluate a big enough sample to be able to draw quantitative conclusions on classification accuracy.

Because questionnaires on journal entry level would take a considerable amount of time, it was chosen to evaluate the outcomes in person. The auditors were not asked to give a definitive judgment on each journal entry, but to scroll trough a large amount of entries and give their general impression

on whether these entries were correctly booked or not. The argumentation results were presented in 4 categories, which are shown in table 4.

	Clustering	Argumentation
Category 1	Not external hire	External hire
Category 2	Not external hire	Not external hire
Category 3	External hire	External hire
Category 4	External hire	Not external hire

Table 4: Categories for evaluation

Furthermore, the evaluation of the journal entries in this way was done twice. The first time the journal entries were presented in a random order, the second time they were sorted by applicable arguments. The experts were asked whether the arguments provided useful insights, and whether it was easier to judge the journal entries when they were presented in groups that had the same arguments apply.

4 Results

In this chapter all results are presented. The results are organized in four sections, corresponding to the four sub-questions.

4.1 Interview results

The first part of the interview was about the current use of data-science in the financial audit and what risks accountants see to the use of data-science. All respondents stated that it is very important for accountants to be able to audit where the data comes from. If they want to rely on some data-analysis tool, their first concern is whether they can be sure the underlying data set is correct and complete. If that cannot be guaranteed, the results of the analysis can only be used to provide an avenue for further search, but never to give assurances.

The second concern accountants had was about the analysis tools themselves. The accountants wanted some way to understand what the tool is doing, and some way to check that the tool does what it promises. Black-box algorithms were considered with suspicion. All respondents indicated that ideally an accountant would understand precisely how the tool functions, but that that is not realistic. They would settle for some way to easily interpret the results, so that it becomes doable for the accountant to judge how much they can rely on the results.

Finally, with respect to the specific problem of finding mistakes in the external hiring journal entries, all respondents note that it is very difficult to judge whether some journal entry is about external hiring based solely on the information available in the general ledger. They indicate that to be sure in difficult cases, that is, the cases where outsourcing and external hire appear most similar, you would need to review the precise wording in the contracts.

4.2 Questionnaire results

In the questionnaires that collected the arguments, a total of 88 distinct arguments were given. However, some arguments were so similar that we have combined them into one argument. For example the fact that the creditor was an employment agency was mentioned as an argument for nine different agencies. These nine distinct arguments were treated as one argument. Another example is the argument that the description contains a name and a month. This argument was given three times; once with respect to the description, once with respect to the header text, and once with respect to booking text. These three arguments were treated as one. After such reductions, 71 distinct arguments remained.

In two cases the experts contradicted each other by giving the same premise for an argument but with contradictory conclusions. However, in both these cases, two experts favoured one view, while the contradictory conclusion was favoured by only one expert. The argument was used with as its conclusion the conclusion that was favoured by the majority.

Table 5 shows for all journal entries in the questionnaire how many arguments were given for that entry and how many of those arguments were a counterargument according to definition 3.1. As can be seen from table 5, for 33 journal entries two arguments were given with the same conclusion for both, meaning that no counterarguments were given. For one journal entry, five arguments were given of which four were a counterargument to another argument.

		Total number of given arguments				
		1	2	3	4	5
Number of counterarguments	0	12	33	21	8	1
	1		10	6	1	
	2			3	1	
	3				2	1
	4					1

Table 5: Frequency table of how many arguments the expert give per journal entry

An argument applies to a journal entry, if the journal entry makes the premise of the argument true. For every journal entry in our data set, at least one of the arguments applied to it. Most journal entries had two to five applicable arguments. The complete distribution of arguments can be found in table 6.

Number of applicable arguments	1	2	3	4	5	6	7	8	9	10
Number of journal entries with that many applicable arguments	13508	39554	68229	84334	40661	8957	1480	460	88	3

Table 6: Number of applicable arguments

When using the definition of counterargument given in definition 3.1, of all 71 arguments, 50 were not countered by any argument, while one argument was countered an impressive 12 times. This was the argument that the document text or text header contain a month. While the experts agreed this is indicative of an external hiring journal entry, there were a lot of cases where these fields may contain a month without being related to external hiring. For example the journal entry could be for juridical costs, and the month could refer to the month the trial was held. The complete distribution of the number of counterarguments using definition 3.1 can be found in table 7.

Number of counterarguments	0	1	2	3	12
Frequency	50	10	6	4	1

Table 7: Number of counterarguments using definition 3.1

As was described in section 3.3, there were concerns that the respondents might not explicitly mention all counterarguments because people tend to confirm and not deny, and because people try to minimize repeating themselves. A bit more liberate definition of the notion of counterargument, definition 3.2, was formulated to deal with this possibility.

When reviewing the completed questionnaires, it was confirmed that the respondents indeed tried

to not be too repetitive. There were cases where a specific characteristic of a journal entry was given as an argument, while for other journal entries with the same characteristic the same respondent did not list it as an argument. The distribution of the number of counterarguments using def 3.2 can be found in table 8.

Number of counterarguments	0	1	2	3	7	12	24
Frequency	42	13	6	6	1	1	2

Table 8: Number of counterarguments using definition 3.2

Of the features that the arguments can apply to, the Document text & Header text was the subject of an argument most often, with creditor a good runner up. The complete list of features that are subject of argument can be found in table 9. The complete list of arguments and their counterarguments can be found in appendix B.

Feature	Number of arguments about that feature
Description	9
Creditor	22
Document text & Header text	29
Amount	5
Budgetary chapter	3
Cost center	3

Table 9: Features that are subject of arguments

The argument that was mentioned most often by the experts is the argument that the document text or text header contain a month. The arguments that applied most often to journal entries in the data set are:

- This journal entry is 'not external hire' because 'amount' is 'below €300'
- This journal entry is 'not external hire' because 'description' contains 'material'
- This journal entry is 'not external hire' because 'creditor' is 'not mentioned'

4.3 Results of cluster analysis and argumentation

In this section the results of both the analyzes are presented.

4.3.1 Clustering results

In table 10 detailed results of a typical cluster assignment are shown. In this case, 1332 anomalies were found that might indicate journal entries that are mistakenly booked as external hiring. These were the external hiring journal entries in clusters 0 and 9. 1347 anomalies were found that might

indicate journal entries that are mistakenly not booked as external hiring. These were the not external hiring journal entries in clusters 2,3,4 and 8. Cluster 6 is a special case where contextual anomalies do not seem to be very relevant, since the ratio of external hiring to other journal entries is almost 1:1. It is especially interesting to see whether the addition of arguments can shed some extra light on this cluster.

Cluster	Size cluster	Amount external hiring	Percentage external hiring	Cluster label
0	232,041	1280	0.55	Not external hire
1	10	0	0	Not external hire
2	980	888	90.6	External hire
3	4,491	3660	81.5	External hire
4	1,044	733	70.2	External hire
5	19	0	0	Not external hire
6	166	84	50.6	External hire
7	107	0	0	Not external hire
8	935	822	87.9	External hire
9	17,481	52	0.30	Not external hire

Table 10: Summary of results clustering analysis

4.3.2 Argumentation results

As was explained in section 3.3, two definition of the notion of counterargument were used. When using definition 3.1, 50 arguments have no counterarguments, while with definition 3.2 42 arguments have no counterarguments. This seems like a relatively small effect, but since mostly arguments that are often applicable receive new counterarguments by definition 3.2, the extra counterarguments result in a lot less journal entries for which no conclusion can be drawn. Tables 11 and 12 show how often the argumentation leads to a conclusion in both cases, if only the argumentation is used to classify all journal entries.

No conclusion	Not external Hiring	External Hiring
97,692	133,347	26,235

Table 11: Results of argument analysis using definition 3.1

No conclusion	Not external Hiring	External Hiring
56,415	174,391	26,468

Table 12: Results of argument analysis using definition 3.2

Tables 11 and 12 show that adding extra counterarguments has the effect that the conclusion that a certain journal entry does not concern external hiring can be drawn more often, while it does not have much effect on the number of the times the conclusion external hire can be drawn.

Furthermore we can see that the conclusion that a certain journal entry concerns external hiring is drawn more often than the amount of external hiring journal journal entries in the data set. The

conclusions external hire was drawn on the basis of the argumentation 26,468 times, while there are only 7519 external hiring journal entries in the data set.

More interesting than to see what the argumentation does with the entire data set, is to see which conclusion the argumentation gives for the anomalies found by the clustering in section 4.3.1. Tables 13, 14 and 15 show the results of the argumentation on the anomalies found by the cluster analysis. In the group of anomalous external hire journal entries, i.e. the journal entries that might be wrongly booked as external hire, about 60% was confirmed as anomalous by the argumentation using definition 3.2. From the anomalous other journal entries, only 6% was confirmed as anomalous by the argumentation. This is not surprising, since the group of other journal entries is huge compared to the group of external hire entries. This means that we expect a high amount of false positives when searching for external hire in the other journal entries, and therefore it is expected that only a small amount of the anomalies is confirmed as anomalous by the argumentation.

	No conclusion	Not external hiring	External hiring	Total
Definition 3.1	594	461	277	1332
Definition 3.2	220	835	277	1332

Table 13: Results of argument analysis journal entries that were booked as external hiring, are anomalous and therefore might be mistaken

	No conclusion	Not external hiring	External hiring	Total
Definition 3.1	565	692	90	1347
Definition 3.2	491	778	78	1347

Table 14: Results of argument analysis journal entries that were booked as not external hiring, are anomalous and therefore might be mistaken

	No conclusion	Not external hiring	External hiring	Total
Definition 3.1	91	24	51	166
Definition 3.2	77	24	65	166

Table 15: Results of argument analysis on cluster 6

4.4 Results of expert evaluation

The results of the argumentation on the anomalous journal entries were evaluated by a financial auditor. When evaluating the final results, the expert noted that journal entries that were marked as a possible mistake based on only one or two applicable arguments, were much more often wrongly identified as a mistake then journal entries for which that conclusion was based on three or more arguments.

In section 3.5 it was described that the expert evaluated the anomalies in combination with the conclusion from the argumentation in four categories. The results of this evaluation can be found

in table 16. It can be seen that the analysis was better in recognizing journal entries that were wrongly booked as external hire, than it was in recognizing journal entries that were wrongly booked as something other than external hire. This was expected, since it is easier to find the mistakes in 7,500 journal entries, than it is to find mistakes in 240,000 journal entries. Furthermore, in both cases where the cluster analysis and the argumentation agree, a larger part of the anomalies was confirmed as actual mistake than when compared to the cases where the cluster analysis and the argumentation disagree.

Booked as	Cluster analysis	Argumentation	Percentage of wrongly booked journal entries according to experts
External hiring	Other	External hiring	75%
External hiring	Other	Other	100%
Other	External hiring	External hiring	50%
Other	External hiring	Other	25%

Table 16: Summary of results clustering analysis, example 1

Finally, and perhaps most importantly, the experts confirmed that knowing which arguments were applied to a journal entry made it significantly easier to understand why a certain journal entry was flagged as a possible mistake. Furthermore, it made it easier to check whether its identification as possible mistake was correct. Where the experts that answered the questionnaires indicated it could take quite some time to evaluate a journal entry, the extra information of the arguments made it considerably more efficient to judge whether the journal entry contained a mistake or not.

5 Discussion

In this chapter the results from chapter 4 are used to answer the four sub-questions and the main research question. After the research question is answered, possible limitations of this research are discussed.

5.1 Answering of the research question

Before answering the research question, the main research question and the sub-questions are recalled:

Can argument-based machine learning and clustering be combined so that they complement each other and thereby improve the use of clustering in an audit setting?

1. What would constitute improvement of the use of clustering for the auditors themselves?
2. What domain arguments and counterarguments do the auditors think should be included in a model to find external hiring entries that might contain mistakes?
3. How do the results of a clustering model change when it is combined with argumentation?
4. Does a model that uses argumentation meet the improvements the auditors formulated?

The first sub-question is very difficult to answer. The main conclusion that can be drawn from the interviews is that while the auditors think it very important that they can interpret and judge the results from data analysis themselves, they do not know precisely what they need and how data-analysis can provide for that need. Because of that, they are incapable of formulating clearly what an improvement to the use of clustering would be for them.

As was discussed in section 2.2, the literature on clustering in audit does provide two ways in which the use of in audit can be improved. The first is mentioned by Torgo et al. [21, 23]. They state that the use of clustering could be improved by a ranking of the found anomalies. They want to be able to rank the anomalies from more likely to be an actual mistake to less likely to be an actual mistake. The second way the use of clustering in audit could be improved is mentioned by Ghani et al. [10]. They state that the evaluation of the anomalies found by the clustering could be made more efficient if the clusters show groups of journal entries that are likely mistaken for similar reasons.

The second sub-question is answered simply by the results that were discussed in section 4.2 and are completely shown in appendix B.

The third sub-question is answered by concluding that the argumentation does not change the clustering results in itself, but is capable of adding information to the clustering. The added information is that the argumentation can confirm or deny that a journal entry is anomalous.

Furthermore, the argumentation can show whether the different clusters have a different distribution of the arguments applied to them.

The final sub-question is answered positively. The ranking in anomalies can be provided by the argumentation, since the expert attested that cases where the argumentation confirmed that a journal entry was anomalous were more often actual mistakes than cases where the argumentation denied that a journal entry was anomalous. Furthermore, the expert confirmed that journal entries that had the same arguments apply to them, were indeed correct or mistaken for similar reasons. Therefore, sorting the anomalies by arguments can indeed improve the efficiency by which the auditor can evaluate the anomalies.

Together the answers to the sub-questions lead to a positive answer to the main research question.

5.2 Limitations of the research

The main limitations of this research are the criteria by which improvement is defined and the implementation of the clustering and the argumentation.

Because the auditors did not have sufficient knowledge of data analysis to formulate what they would see as an improvement, it proved very difficult to pinpoint what exactly counts as an improvement, thereby making the criteria by which these improvements could be measured unclear. Added difficulty was that there was no sample of which it was known whether the journal entries were correctly booked or not. Because of this it was impossible to measure improvement quantitatively.

Another limitation is the unconventional choices that were made in the data transformation and the implementation of the clustering. Because of this, the results of the clustering cannot be validated based on existing literature. However, since the main finding of the research is that argumentation can add useful extra information to cluster results, this finding still stands if the clustering in this research is replaced by a more conventional way of clustering.

Finally, the ad hoc method used to implement the argumentation in this research is a limitation. Since no existing method was used, validation based on the existing literature was again not possible.

6 Conclusion

The research that was carried out in this thesis can be summarized as follows: Interviews and questionnaires were conducted to ascertain what improvement of clustering would constitute, and what arguments needed to be used in a model. Next, k-means clustering was implemented and the anomalies that were found with the clustering were subjected to argumentation. Finally, the anomalies enriched with the information from the arguments were evaluated by a financial auditor, confirming that the argumentation was of added value to the auditors. Thereby the main research question, "Can argument-based machine learning and clustering be combined so that they complement each other and thereby improve the use of clustering in an audit setting?", was answered in the positive.

Recommendations for future work
There is room for improvement in the collection of the arguments. First, it was noted by the experts that journal entries that had only one or two applicable arguments were way more often misidentified on the basis of the argumentation than journal entries that had three or more applicable arguments. Furthermore, specific journal entries showed that there was reason to assume that to little arguments were used. For example, 20 entries in the set of anomalous external hire journal entries concern the hiring of a car and chauffeur for a specific state visit. The experts agree that this is not external hire but outsourcing, since it is a product that is bought (transportation for a visiting queen on a specific day). In the questionnaires both the terms 'driver' and 'limousine' were given as reasons to think journal entries were not external hire. However, the specific description of 'car with chauffeur' did not occur in the questionnaires, and therefore there is no argument that covers it. Of course it is reasonable to assume that if drivers of limousines are not external hire, then neither are chauffeurs for cars. Based on the fact that experts indicate that journal entries with little arguments are more often misclassified and that some journal entries should be covered by the arguments but are not because of specific wording, it would be desirable to do a second round to collect arguments from experts, where previous arguments can be expanded or narrowed to make sure the arguments cover the relevant cases as precisely as possible. This should minimize the cases where only a few arguments are applicable, thereby making the argumentation more valuable.

Another aspect of the argument collection that is open for improvement is that of the counterarguments. As we have discussed in section 4.2, people tend to corroborate and not falsify, so despite the encouragement that counterarguments were important for the research, little counterarguments were provided. This gives an argument like "there is a month mentioned in the description" to much force. There are enough cases conceivable where a month in the description is countered by some other element of the journal entry.

Possible avenues for further research are to combine clustering and argumentation as in this research, but to use a different clustering technique. It could also prove interesting to conduct a similar research on a labeled data set, so that the added value of the argumentation can be quantified.

References

[1] Mohiuddin Ahmed, Abdun Naser Mahmood, and Md Rafiqul Islam. "A survey of anomaly detection techniques in financial domain". In: *Future Generation Computer Systems* 55 (2016), pp. 278–288.

[2] Leila Amgoud and Mathieu Serrurier. "Agents that argue and explain classifications". In: *Autonomous Agents and Multi-Agent Systems* 16.2 (2008), pp. 187–209.

[3] Floris J Bex. *Arguments, stories and criminal evidence: A formal hybrid theory*. Vol. 92. Springer Science & Business Media, 2011.

[4] Lucas Carstens and Francesca Toni. "Improving out-of-domain sentiment polarity classification using argumentation". In: *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*. IEEE. 2015, pp. 1294–1301.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[6] Oana Cocarascu and Francesca Toni. "Argumentation for Machine Learning: A Survey." In: *COMMA*. 2016, pp. 219–230.

[7] Phan Minh Dung. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games". In: *Artificial intelligence* 77.2 (1995), pp. 321–357.

[8] Yang Gao and Francesca Toni. "Argumentation Accelerated Reinforcement Learning for Cooperative Multi-Agent Systems." In: *ECAI*. 2014, pp. 333–338.

[9] Alejandro Javier Garcıa and Guillermo Ricardo Simari. "Defeasible Logic Programming: An Argumentative Approach". In: *CoRR* cs.AI/0302029 (2003).

[10] Rayid Ghani and Mohit Kumar. "Interactive learning for efficiently detecting errors in insurance claims". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 325–333.

[11] Sergio Alejandro Gómez and Carlos Iván Chesnevar. "Integrating defeasible argumentation with fuzzy art neural networks for pattern classification". In: *Journal of Computer Science & Technology, 2004, vol. 4, núm. 1, p. 45-51* (2004).

[12] Sergio Alejandro Gómez and Carlos Iván Chesñevar. "Integrating Defeasible Argumentation and Machine Learning Techniques". In: *CoRR* cs.AI/0402057 (2004).

[13] Data Analytics Working Group. *Exploring the Growing Use of Technology in the Audit, with a Focus on Data Analytics*. 2016. URL: `https://www.ifac.org/publications-resources/exploring-growing-use-technology-audit-focus-data-analytics`.

[14] H2O.ai. *R Interface for H2O*. R package version 3.10.0.8. Oct. 2016. URL: `https://github.com/h2oai/h2o-3`.

[15] Mieke Jans, Nadine Lybaert, and Koen Vanhoof. "Data mining for fraud detection: Toward an improvement on internal control systems?" In: *European Accounting Association - Annual Congress, 30, Lisbon* (2007).

[16] Robert Koons. "Defeasible Reasoning". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2017. Metaphysics Research Lab, Stanford University, 2017.

[17] Martin Možina, Jure Žabkar, and Ivan Bratko. "Argument based machine learning". In: *Artificial Intelligence* 171.10 (2007). Argumentation in Artificial Intelligence, pp. 922–937.

[18] Krystyna Napierała and Jerzy Stefanowski. "Argument Based Generalization of MODLEM Rule Induction Algorithm". In: *Rough Sets and Current Trends in Computing*. Ed. by Marcin Szczuka et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 138–147.

[19] Santiago Ontañón et al. "A defeasible reasoning model of inductive concept learning from examples and communication". In: *Artificial intelligence* 193 (2012), pp. 129–148.

[20] Andrei Sorin Sabau. "Survey of clustering based financial fraud detection research". In: *Informatica Economica* 16.1 (2012), p. 110.

[21] C Soares and R Ghani. "Resource-bounded outlier detection using clustering methods". In: *Data Mining for Business Applications* 218 (2010), p. 84.

[22] Sutapat Thiprungsri and Miklos A Vasarhelyi. "Cluster analysis for anomaly detection in accounting data: An audit approach". In: *The International Journal of Digital Accounting Research* (2011).

[23] Luis Torgo and Elsa Lopes. "Utility-based fraud detection". In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. Vol. 22. 1. 2011, p. 1517.

[24] Anthony K. H. Tung. "Rule-based Classification". In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 2459–2462.

[25] Joost Vandewal and Ir Remco Dijkman–TU. "Towards a more efficient audit process: A data-driven approach". In: *Thesis Archive Technische Universiteit Eindhoven* (2016).

[26] Miklos A Vasarhelyi and Hussein Issa. *Application of anomaly detection techniques to identify fraudulent refunds*. Tech. rep. Working Paper. Rutgers Business School, Rutgers Accounting Research Center, 2011.

[27] Sankar Virdhagriswaran and Gordon Dakin. "Camouflaged fraud detection in domains with complex relationships". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2006, pp. 941–947.

A Interviews

A.1 Interview questionnaire

Introduction

- About myself
- About my thesis project
  - Explain goals of the project
  - Explain what clustering and formal argumentation are
  - Explain the role of this interview in the project
- Discuss length of interview
- Discuss subject that will be covered in the interview
- Discuss agreements
  - Recording the interview
  - Sending the transcript
  - Sending the final thesis

Function of the respondent in the organization

- What is your function?
- What is your background?
- What are your responsibilities?
- ...

Current practices and standards

- Could you give an example of how an average financial control works?
- What is data-science according to you?
- Do you use data-science in the financial control?
  - If so, how?
  - If not, why not?
- Do you see more ways in which data-science could be useful in the financial control?

- What risks do you see for the use of data-science?
- What do you think about the use of black-box algorithms for the financial audit?
- According to literature, accountants are reserved with using black-box algorithms. Do you recognize this?
- ...

External hiring journal entries The problem case for my thesis is to identify external hiring journal entries by means of clustering.

- Do you think data-science could be useful with respect to this problem?
- How does the audit of external hiring work right now?
- What are characteristics of a journal entry that are relevant for finding external hiring entries?
- And what are parts of a journal entry are not relevant?

For the first model I have made some assumptions.

- I have used week number instead of date. Do you think this is a logical choice?
- I have used only the first two digits of the general ledger account numbers. Do you think this is a logical choice?
- What could possible consequences of these simplifications be?

Argumentation The goal is to improve the model by adding arguments that experts use when reasoning about this problem.

- Can you give arguments that you would use to argue that a specific journal entry concerns external hiring?
- Can you give arguments that you would use to argue that a specific journal entry does not concern external hiring?
- Can you give arguments that you would use to argue that a specific journal entry about external hiring concerns a mistake.
- Can you order these arguments? Which arguments are more persuasive than others?

Conclusion

- Shortly summarize the interview

- Check if all relevant issues have been discussed
- Repeat agreements made
- Ask what respondent thought of interview
- Thanking the respondent

B Arguments

Argument	Feature	is	Conclusie	Tegenargumenten
1	Omschrijving	Loonkosten lokaal personeel	Geen inhuur	10
2	Omschrijving	HPO variabele personele uitgaven	Inhuur	11, 14, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 60
3	Omschrijving	Bevat HPO of HDPO	Inhuur	
4	Omschrijving	leeg	Geen inhuur	
5	Omschrijving	Materieel	Geen inhuur	10, 24
6	Omschrijving	niet "Lokaal personeel"	Inhuur	
7	Omschrijving	Inhuur / External Hire	Inhuur	
8	Creditnaam	Inquest	Inhuur	14
9	Omschrijving	Diensverleningsovk	Inhuur	14
10	Creditnaam	Uitzendbureau	Inhuur	62, 68
11	Creditnaam	Advocatenbureau	Geen inhuur	
12	Creditnaam	Eigennaam	Inhuur	1, 32, 42
13	Creditnaam	Niet ingevuld	Geen inhuur	
14	Creditnaam	Ander ministerie, gemeente of ambassade	Geen inhuur	
15	Creditnaam	TNT	Geen inhuur	
16	Creditnaam	Atos	Geen inhuur	
17	Creditnaam	Limousines, taxibedrijf	Geen inhuur	
18	Creditnaam	Fire & security	Geen inhuur	
19	Creditnaam	Adviesgroep	Geen inhuur	
20	Creditnaam	Language Courses	Geen inhuur	
21	Creditnaam	Publishers	Geen inhuur	
22	Creditnaam	KSG	Geen inhuur	
23	Creditnaam	VX company IT service	Geen inhuur	
24	Creditnaam	People Performance	Inhuur	1
25	Creditnaam	Recruitement	Inhuur	1
26	Creditnaam	Pbp public partners	Inhuur	1
27	Creditnaam	VFS global services	Inhuur	1
28	Creditnaam	Admb Afwezigheidscontrole	Inhuur	1
29	Creditnaam	Presenter Utrecht	Inhuur	1
30	Creditnaam	bevat "web"	Inhuur	1
31	Creditnaam	Technisch Bureau koppes	Inhuur	1, 4, 38
32	Tekst boeking of Doc koptekst	Report	Geen inhuur	
33	Tekst boeking of Doc koptekst	Transportkosten	Geen inhuur	

34	Tekst boeking of Doc koptekst	Dashboard	Geen inhuur	
35	Tekst boeking of Doc koptekst	Koerier	Geen inhuur	
36	Tekst boeking of Doc koptekst	NVIS	Geen inhuur	
37	Tekst boeking of Doc koptekst	Prive	Geen inhuur	
38	Tekst boeking of Doc koptekst	leeg	Geen inhuur	10, 26, 61
39	Tekst boeking of Doc koptekst	Daily meal	Geen inhuur	
40	Tekst boeking of Doc koptekst	Extra driver	Geen inhuur	
41	Tekst boeking of Doc koptekst	Vacatiegelden	Geen inhuur	
42	Tekst boeking of Doc koptekst	Cleaning lady, schoonmaakkosten	Geen inhuur	
43	Tekst boeking of Doc koptekst	Beleidsdoorlichting	Geen inhuur	
44	Tekst boeking of Doc koptekst	DJZ, Staat, juridische kosten	Geen inhuur	
45	Tekst boeking of Doc koptekst	IOB evaluatie	Geen inhuur	
46	Tekst boeking of Doc koptekst	Lokaal pers	Geen inhuur	
47	Tekst boeking of Doc koptekst	Temporary Staff	Geen inhuur	
48	Tekst boeking of Doc koptekst	Car rent	Geen inhuur	
49	Tekst boeking of Doc koptekst	Sharepoint	Geen inhuur	
50	Tekst boeking of Doc koptekst	Medical costs	Geen inhuur	
51	Tekst boeking of Doc koptekst	Les	Geen inhuur	
52	Tekst boeking of Doc koptekst	Advies verlichting	Geen inhuur	
53	Tekst boeking of Doc koptekst	Inhuur / External Hire	Inhuur	62, 68
54	Tekst boeking of Doc koptekst	Uren	Inhuur	1, 14

55	Tekst boeking of Doc koptekst	Eigennaam en maand	Inhuur	1, 5, 11, 14, 15, 22, 35, 42, 44, 50, 63, 69
56	Tekst boeking of Doc koptekst	Advies	Inhuur	1, 11, 14
57	Tekst boeking of Doc koptekst	Ext Staff	Inhuur	1, 14
58	Tekst boeking of Doc koptekst	Bevat HPO of HDPO	Inhuur	14, 60
59	Tekst boeking of Doc koptekst	Bevat "vac" voor vacature	Inhuur	14
60	Bedrag	Lager dan 300	Geen inhuur	
61	Bedrag	Tussen 750 en 2500	Inhuur	1, 5, 11, 41, 44, 45, 52
62	Bedrag	Hoger dan 15.000	Geen inhuur	55
63	Bedrag	Hoger dan 8000	Geen inhuur	
64	Bedrag	afgerond bedrag	Geen inhuur	2, 3, 55
65	Operationele Doelstelling	0504U02-Consulaire dienstverlening vreemdelingen	Geen inhuur	
66	Operationele Doelstelling	0504u01-Consulaire dienstverlening NL in buitenl	Geen inhuur	
67	Operationele Doelstelling	niet Apparaat	Geen inhuur	54, 55, 61
68	Documentsoort	WE-Goederenontvangst	Geen inhuur	
69	Kostenplaats	HPO	Inhuur	11, 14, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 60
70	Kostenplaats	YAN	Geen inhuur	
71	Kostenplaats	DBV	Geen inhuur	31