FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

FACHBEREICH WIRTSCHAFTS-
WISSENSCHAFTEN

Unregelmäßigkeiten in der Abschlussprüfung – Fraud Detection mit Hilfe von Machine Learning

Bachelorarbeit

Eingereicht von:	Marco Döll
	Tannenstraße 5a
	90552 Röthenbach an der Pegnitz
Matrikelnummer:	22259692
Studiengang:	Wirtschaftsinformatik
Referent:	Prof. Dr. Michael Amberg
Betreuer:	Oleg Seifert
Bearbeitungszeit:	31.07.2020- 28.09.2020

Abstract

Das Ziel dieser Forschung ist es zu untersuchen, inwieweit der Betrugserkennung in der Wirtschaftsprüfung durch verschiedene Analysen assistiert werden kann. Dazu wird die folgende Forschungsfrage gestellt: Wie kann der Abschlussprüfer mit Hilfe von Machine Learning dabei unterstützt werden, Betrugsfälle aufzudecken? Um Zu ihrer Beantwortung wurde zum einen eine strukturierte Literaturanalyse durchgeführt, zum anderen unterschiedliche Algorithmen anhand zweier Datensätzen und einem ausgewählten Anwendungsfall verprobt sowie mittels Metriken zur Evaluierung miteinander verglichen. Die Ergebnisse der Untersuchung zeigen, dass Auffälligkeiten in den Datensets identifiziert und Hinweise an den Abschlussprüfer übermittelt werden können, welche im Anschluss zur Aufdeckung von Betrugsfällen beitragen.

Basierend auf den Ergebnissen der Forschung ist es empfehlenswert, Analysen mit Hilfe von Machine Learning für die Erkennung von Auffälligkeiten in der Abschlussprüfung einzusetzen. Jedoch sollten weitere Anwendungsfälle sowie Datensätze für zukünftige Untersuchungen in Betracht gezogen werden.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AD	Anomaly Detection
AUC	Area under the Curve
EM	Excess Mass
FP	False Positives
FN	False Negatives
FF	Forschungsfrage
IDW	Institut der Wirtschaftsprüfer
JET	Journal Entry Testing
LF	Leitfrage
LOF	Local Outlier Factor
MV	Mass Volume
OD	Outlier Detection
PwC	PricewaterhouseCoopers
ROC	Receiver operating characteristics
SVM	Support Vektor Maschine
TP	True Positives
TN	True Negatives
TF	Teilfrage

1. Einleitung

1.1 Einleitung und Motivation

Eine in 2017 von PricewaterhouseCoopers (PwC) in Zusammenarbeit mit der Martin-Luther-Universität Halle-Wittenberg durchgeführte Studie mit 500 Unternehmen aus verschiedenen Wirtschaftsbereichen ergab, dass 45% der Firmen innerhalb der letzten zwei Jahre von Wirtschaftskriminalität betroffen waren. Neben dem starken Anstieg von Cyberkriminalität stellen vor allem Vermögensdelikte und Falschbilanzierungen eine häufige Form von analoger Wirtschaftskriminalität dar. (Bussmann, Nestler, & Salvenmoser, 2018)

Die finanziellen Auswirkungen von Wirtschaftskriminalität lassen sich nur bedingt beziffern, da indirekte Folgen wie „Schadenersatzforderungen, straf-bzw. bußgeldrechtliche Haftungsrisiken, Kosten für Rechtsverfahren, Stakeholdermanagement und Reputationsschäden“ (Bussmann, Nestler, & Salvenmoser, 2018, S. 21) zu berücksichtigen sind. Der durchschnittliche Schaden pro Delikt beläuft sich auf circa 723.000€, wobei größere Unternehmen deutlich höhere Schäden zu verzeichnen haben. Darüber hinaus beträgt der mittlere Verlust bei besonders schwerwiegenden Delikten 7,23 Millionen Euro. (Bussmann, Nestler, & Salvenmoser, 2018)

Zunehmend sind Beispiele von Finanzbetrug in den Medien, wie die Unterschlagung von 100 Millionen Dollar bei einer koreanischen Tochter des Schweizer Industriekonzerns ABB. Daher wird immer wieder die Frage aufgeworfen, welche Verantwortung der Prüfer hierbei trägt. (Schmitt, 2017) In den vom Institut der Wirtschaftsprüfer (IDW) publizierten Prüfungsstandards wird vom Abschlussprüfer verlangt, im Rahmen der Abschlussprüfung wesentliche Falschangaben mit hinreichender Sicherheit festzustellen. (Institut der Wirtschaftsprüfer, 2006)

Klaus Peter Neumann, Vorstandsmitglied des IDW nimmt hierzu Stellung: „Die Abschlussprüfung ist aber nicht speziell darauf ausgerichtet, kriminelle Handlungen aufzudecken.“ (Schmitt, 2017, S. 1) Der Abschlussprüfer geht dabei mit einer kritischen Grundhaltung bei der Prüfung vor. Solange keine Hinweise oder Anhaltspunkte vorliegen, muss sich der Abschlussprüfer jedoch auf die Authentizität, der ihm vom Unternehmen bereitgestellten Unterlagen und Dokumente verlassen. (Schmitt, 2017)

Diese Bachelorarbeit fokussiert sich auf die zuvor beschriebene Problematik. Da die Wirtschaftsprüfungskanzleien oftmals hohem zeitlichen und personellen Druck ausgesetzt sind, soll im Rahmen dieser Arbeit untersucht werden, ob und welche

Analysen zur Erkennung von Betrügen, als sinnvolle und ergänzende Hilfsmittel in Betracht gezogen werden können.

1.2 Zielsetzung und Forschungsfragen

Das Ziel der Forschung ist es, herauszufinden, inwiefern die tägliche Arbeit der Abschlussprüfer bezüglich der Betrugserkennung mit Hilfe von Machine Learning unterstützt werden kann. Hieraus leitet sich die erste Forschungsfrage (FF) ab:

- **FF)** Wie kann der Abschlussprüfer mit Hilfe von Machine Learning dabei unterstützt werden, Betrugsfälle aufzudecken?

Um die Leitfrage in ihrer Komplexität beantworten zu können, ist es nötig diese in weitere Teilfragen (TF) zu unterteilen:

- **TF1)** Welche Methoden der Ausreißererkennung eignen sich für den Einsatz der Betrugserkennung in der Abschlussprüfung?

- **TF2)** Wie können die Ergebnisse der Datenanalyse evaluiert werden?

- **TF3)** Wie können Machine Learning Algorithmen zur Betrugserkennung in der Abschlussprüfung angewandt werden?

Zunächst gilt es, mit Hilfe einer strukturierte Literaturanalyse, geeignete Techniken und Konzepte zur Betrugserkennung und Evaluierung für den Sachverhalt zu identifizieren. Diese Analyse erfolgt im Rahmen der Teilfragen TF1 und TF2.

Anschließend werden im Zuge der Frage TF3 die in der Literaturanalyse beschriebenen Methoden und Konzepte an zwei wesentlichen Datensätzen der Abschlussprüfung angewandt sowie anhand verschiedener Metriken zur Evaluierung miteinander verglichen.

1.3 Forschungsdesign

Das Vorgehen zur Beantwortung der Forschungsfragen ist an dem von Hevner et al (2004) beschriebenen Ansatz des Design Science angelehnt. Die grundsätzliche Idee besteht darin, die Forschung durch das Konstruieren und Evaluieren von sogenannten Artefakten den identifizierten Geschäftsanforderungen anzupassen.

Darüber hinaus orientiert sich die Methodik im Rahmen dieser Bachelorarbeit an dem von Pfeffers et al (2007) beschriebenen Vorgehensmodell der Design Science Research. Dieser Ansatz wurde für die vorliegende Untersuchung angepasst und ist in folgender Tabelle dargestellt.

Prozesselemente nach Pfeffers et al (2007)	Umsetzung in der Arbeit	Ergebnisse
Identify Problem & Motivate	Problemstellung anhand verschiedener Beispiele von Finanzbetrugsfällen.	Motivationsbeschreibung (Kapitel 1)
Design & Development	Literaturanalyse zur Sammlung von Methoden der Ausreißererkennung und Metriken zur Evaluierung.	Analysen von relevanten Methoden und Ansätzen (Kapitel 3)
Demonstration	Einsatz des Vorgehensmodells anhand zwei unterschiedlicher Datensets.	Dokumentation des Vorgehensmodells und der Implementierung (Kapitel 4)
Evaluation	Anwendung verschiedener Metriken zur Überprüfung der Analyseergebnisse.	Vorstellung und Interpretation der Ergebnisse der Untersuchung (Kapitel 5-6)
Communication	Veröffentlichung der Ergebnisse in der Bachelorarbeit.	Bachelorarbeit

Tabelle 1: Vorgehensweise der Arbeit in Anlehnung an dem Design Science Research Prozess von Pfeffers et al (2007)

1.4 Aufbau der Arbeit

Im ersten Teil verschafft die Einleitung einen Überblick über den Themenbereich und soll den Einstieg in das Thema vereinfachen. Zusätzlich werden die Zielsetzung sowie die in der Arbeit zu beantwortenden Forschungsfragen definiert. Darüber hinaus wird das Forschungsdesign beschrieben.

Anschließend behandelt das zweite Kapitel den der Thesis zugrunde liegenden theoretischen Hintergrund. Hier wird besonders auf unterschiedliche Definitionen von Begriffen eingegangen, die in den darauffolgenden Abschnitten verwendet werden und zum Verständnis der Arbeit unerlässlich sind.

Der dritte Teil untersucht in Form einer Literaturanalyse die Forschungsfragen TF1 und TF2 und geht hierbei insbesondere auf verschiedene Methoden der Ausreißererkennung und Konzepte zur Evaluierung ein.

Daraufhin werden im vierten Abschnitt auf Basis der, in der Literaturanalyse vorgestellten theoretischen Modelle, drei ausgewählte Algorithmen an zwei Datensätzen der Wirtschaftsprüfung verprobt. Zusätzlich dienen bereits existierende Prozesse und Analysen in der Wirtschaftsprüfung der Identifikation eines geeigneten Anwendungsfalles für die Untersuchung der Daten.

Ferner sollen die Ergebnisse des vorangehenden Abschnittes im fünften Kapitel anhand geeigneter Metriken zur Evaluierung verglichen und diskutiert werden.

Mit Hilfe der Kapitel vier und fünf wird die Forschungsfrage TF3 beantwortet.

Abschließend wird die Arbeit zusammengefasst und ein kurzer Ausblick in die Zukunft des Themengebiets gegeben.

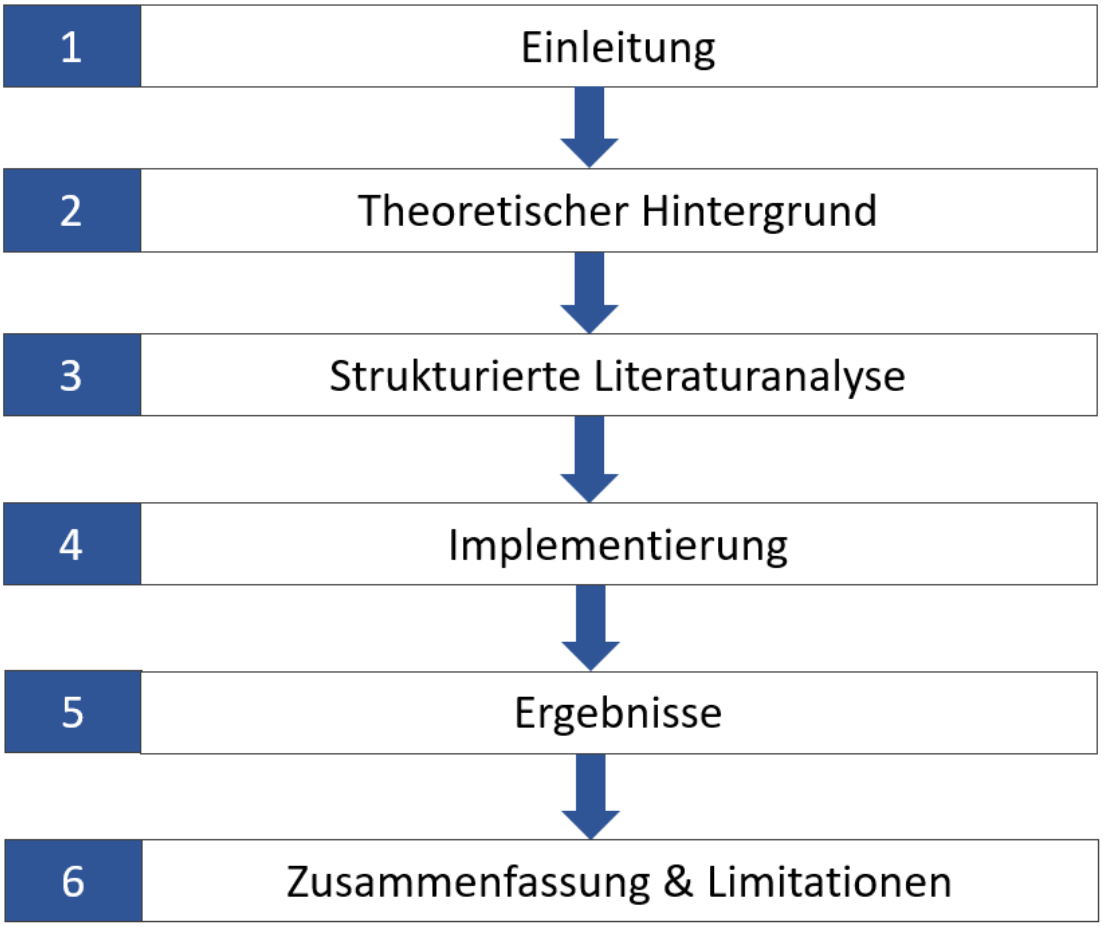


Abbildung 1: Aufbau der Arbeit

2. Theoretischer Hintergrund

Dieses Kapitel vermittelt einen Überblick über den theoretischen, der Arbeit zugrundeliegenden Hintergrund. Zunächst wird der Begriff des Data Minings vom maschinellen Lernen (engl.: Machine Learning) abgegrenzt. Darüber hinaus wird der Unterschied zwischen überwachtem und unüberwachtem maschinellem Lernen dargestellt. Anschließend werden die Bezeichnungen Ausreißer und Fraud Detection definiert.

2.1 Data Mining

Hand et al (2015) beschreiben Data Mining als die Technologie zur Erkennung von Strukturen und Mustern in großen Datensets. Sie legen dar, dass Data Mining eine wesentliche Überlappung mit anderen Disziplinen der Datenanalyse, wie beispielsweise Statistik, maschinellem Lernen und Mustererkennung besitzt. Dennoch kann Data Mining aufgrund der Größe der Datensets, der oftmals schlechten Qualität der Daten sowie der Vielfalt der gesuchten Strukturen von den anderen Disziplinen abgegrenzt werden.

Auch Han et al (2012) sehen Data Mining als interdisziplinäres Feld an und zeigen auf, dass der Begriff auf viele verschiedene Weisen definiert werden kann. Ferner bevorzugen sie den Begriff des „knowledge mining from data“ und spezifizieren den Terminus als „Suche nach Wissen in Daten“ (Han, Kamber, & Pei, 2012, S. 6).

2.2 Überwachtes und unüberwachtes maschinelles Lernen

Wie bereits im vorangehenden Kapitel erläutert, überlappen sich die Begriffe maschinelles Lernen und Data Mining wesentlich. Reitmaier (2015) beschreibt, dass maschinelles Lernen „das Ziel der Generalisierung verfolgt, d.h. ausgehend von Beispieldaten Muster der Gesetzmäßigkeiten einer Problemstellung zu erlernen bzw. zu verallgemeinern, um somit auch unbekannte Daten der gleichen Problemstellung beurteilen zu können.“ (Reitmaier, 2015, S. 240)

Zusätzlich legen Müller et al (2017) dar, dass maschinelles Lernen darauf abzielt, Wissen aus Daten zu extrahieren. Sie verdeutlichen darüber hinaus, dass Anwendungen von Methoden des maschinellen Lernens in den letzten Jahren in unseren Alltag integriert wurden. Dabei verweisen sie beispielsweise auf automatische Empfehlungen für Filme, Nahrungsmittel sowie personalisierte Werbung.

Im Allgemeinen definieren Mohri et al (2012) maschinelles Lernen als analytische Methode, die Erfahrungen beziehungsweise historische Daten verwendet, um die Leistung von Anwendungen zu optimieren oder möglichst genaue Vorhersagen zu treffen.

Maschinelles Lernen kann in vier Kategorie eingeteilt werden: überwachtes-, unüberwachtes-, teilüberwachtes und verstärkendes Lernen (Alpaydin, 2009). Im Rahmen dieser Arbeit wird lediglich auf unüberwachtes- sowie überwachtes Lernen eingegangen.

Beim überwachten Lernen empfängt das Lernmodell „ein Satz gelabelter Beispiele als Trainingsdaten und macht Vorhersagen für alle ungesehenen Punkte“. (Mohri, Rostamizadeh, & Talwalkar, 2012, S. 7)

Als gelabelte Daten werden Datensätze beschrieben, die mit einem Label annotiert wurden (cloudfactory, 2020). So werden den einzelnen Datenpunkten Labels zugeordnet, welche diese in eine bestimmte Kategorie einordnen (Google LLC, 2020). Beispielweise werden in einem binären Klassifizierungsproblemen den einzelnen Datenpunkten das Label „Spam“ oder „non-Spam“ zugeordnet.

Beim unüberwachten Lernen hingegen empfängt das Lernmodell ausschließlich ungelabelte Daten und versucht Muster und Vorhersagen im Datensatz zu erkennen (Google LLC, 2020). Darüber hinaus legen Mohri et al (2012) dar, dass die Leistung des Lernmodells bei unüberwachten Lernen sehr schwer evaluiert werden kann, da keinerlei Label präsent sind.

2.3 Ausreißer

Zunächst soll der Begriff Ausreißer, in der Literatur meist unter den englischen Begriffen „Outlier“ oder „Anomaly“ zu finden, definiert werden. Hawkins formuliert in seinem Buch „Identification of Outliers“ aus dem Jahre 1980 eine intuitive Definition der Bezeichnung Outlier: „an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.“ (Hawkins, 1980, S. 1). Viele Artikel und Bücher berufen sich auf die Definition von Hawkins, wie zum Beispiel Han et al (2012) und Pahuja & Yadav (2013). Ausreißer werden auch als Werte definiert, die im Auge des Forschers zweifelhaft sind (Dixon, 1950).

Obwohl unterschiedliche Interpretationen des Begriffs existieren, sind diese dennoch in ihrer generellen Auffassung weitestgehend ähnlich und charakterisieren Ausreißer als in irgendeiner Form verdächtig, zweifelhaft oder nicht normal.

2.4 Fraud Detection

Chandola et al (2009) definieren den Begriff der Fraud Detection (deutsch: Betrugserkennung) als Aufdeckung krimineller Aktivitäten in kommerziellen Organisationen, wie beispielsweise Banken oder Versicherungsunternehmen.

In engem Zusammenhang mit der Erkennung von Betrug steht das Konzept des sogenannten „Fraud Triangle“. Diese Vorgehenweise, im deutschsprachigen Raum bekannt als „Betrugs-Dreieck“, fasst die drei ausschlaggebenden Entstehungsgründe von Wirtschaftskriminalität zusammen. Das Dreieck wurde bereits in den 1940er Jahren von Donald R. Cressey entwickelt. Es legt die Tatsache zugrunde, dass Wirtschaftskriminalität auftreten kann, wenn für den Täter eine Gelegenheit zur Tat besteht, ein gewisser Anreiz oder Druck für die Tat vorhanden ist und eine geringe Moral des Täters nach der Tat vorzuweisen ist. (Hofmann, 2008)

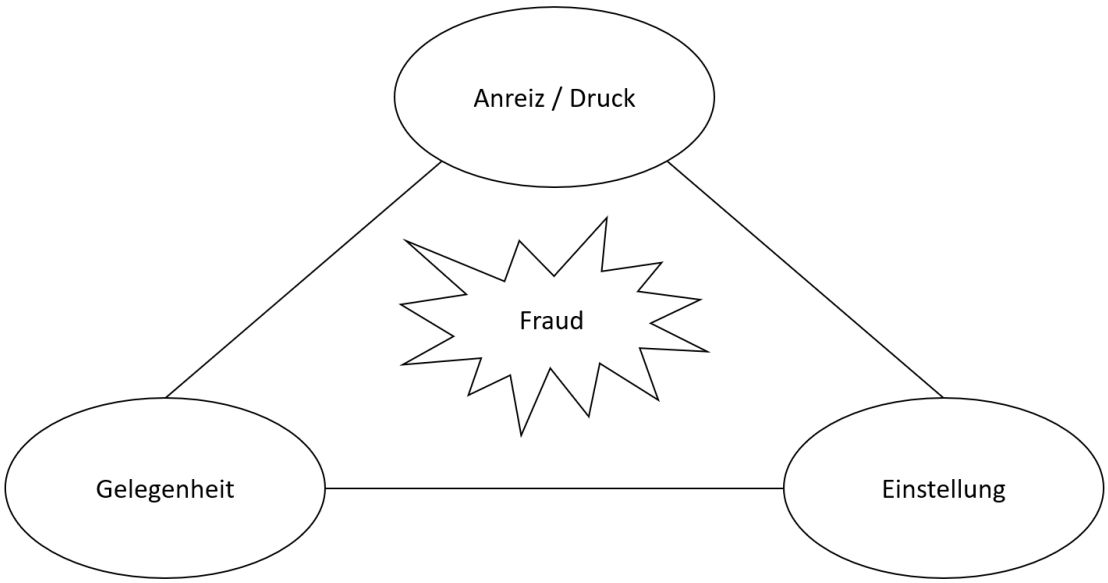


Abbildung 2: Fraud Triangle (in Anlehnung an: (Hofmann, 2008, S. 204))

3. Strukturierte Literaturanalyse

Um relevante Literatur zu identifizieren, wurde ein dreistufiger Prozess angewandt, der an den strukturierten Ansatz von Webster und Watson (2002) angelehnt ist. Zuerst wurden die Keywords: „Outlier Detection“ (OD), „Anomaly Detection“ (AD) sowie „Evaluation“ in Kombination mit OD und AD verwendet, um Artikel in führenden Fachzeitschriften zu ermitteln. Außerdem wurden die eben genannten Schlüsselwörter mit „Financial Fraud“ sowie „Accounting Fraud“ zusammengesetzt, um Nischenpublikationen und Bücher zum speziellen Themengebiet der Betrugserkennung bei Wirtschaftsprüfern zu identifizieren.

Daraufhin wurden die Zitate der bisher gesammelten Publikationen analysiert und weitere, häufig zitierte und relevante Artikel der Literatursammlung hinzugefügt (backward search). Schlussendlich wurden weitere Quellen ergänzt, indem Artikel und Bücher mit Hilfe von Google Scholar gesucht wurden, welche die wichtigsten Veröffentlichungen der vorherigen Schritte zitiert haben (forward search). In den letzten beiden Schritten gab es keine Einschränkung bezüglich des Rankings der Fachzeitschriften. Zudem wurden sowohl Bücher als auch Paper zu noch laufenden Forschungen in Betracht gezogen, um eine ganzheitliche Abdeckung aller relevanten Arbeiten zu ermöglichen. Im Rahmen dieser Arbeit sollen mit Hilfe der Literaturanalyse Teilfrage 1 und 2 beantwortet werden:

- **TF1)** Welche Methoden der Ausreißererkennung eigenen sich für den Einsatz der Betrugserkennung in der Abschlussprüfung?

- **TF2)** Wie können die Ergebnisse der Datenanalyse evaluiert werden?

3.1 Methoden der Ausreißererkennung

In der Literatur und Praxis existieren zahlreiche Methoden der Ausreißererkennung. Diese sind meist unter den englischen Begriffen Outlier Detection und Anomaly Detection zu finden. Im folgenden Kapitel werden hierzu sechs Methoden und jeweils ein Beispiel für Algorithmen des Ansatzes vorgestellt. Im Rahmen dieser Arbeit wird lediglich auf einen Teil der Methoden der Ausreißererkennung eingegangen, da es nicht das Ziel der Literaturanalyse ist, jedes existierende Konzept sowie Algorithmus aufzulisten. Folglich erhebt die Untersuchung keinen Anspruch auf Vollständigkeit.

3.1.1 Statistische Methoden

Statistische Ansätze tauchen bereits sehr früh in der Literatur auf und stellen das früheste Konzept zur Erkennung von Ausreißern dar. So beschreiben sowohl Rousseeuw und Leroy (1987) als auch Barnett und Lewis (1995) Techniken, die vor allem für eindimensionale Probleme zum Einsatz kommen. Darüber hinaus heben Hodge und Austin (2004) hervor, dass statistische Modelle generell für quantitative, reelle Datensets geeignet sind und deren Bearbeitungszeit erhöht wird, sobald komplexe Datentransformationen vor der Verarbeitung durchgeführt werden müssen.

In klassischen Software-Anwendungen für die Datenanalyse bei Wirtschaftsprüfern wie beispielsweise „IDEA“ der Audicon GmbH oder dem Produkt „Digitale Datenanalyse“ der DATEV eG finden sich Analyseschritte zum Thema Benford's Gesetz. Auch in der Literatur häufen sich die Artikel über dieses Thema. Es definiert die Häufigkeitsverteilung von Ziffern in einem Datensatz ab der ersten bis zur vierten Position, von links beginnend (Tammaru & Alver, 2016). Ferner legen Durtschi et al (2004) dar, dass ein großer Teil der Veröffentlichungen „die Anwendung dieses Gesetzes als einfache und effektive Möglichkeit für Wirtschaftsprüfer beworben haben, um sowohl betriebliche Unstimmigkeiten festzustellen als auch Betrug bei den Rechnungslegungszahlen aufzudecken.“ (Durtschi, Hillison, & Pacini, 2004, S. 1)

Dennoch hegen auch einige Forscher Zweifel an der Effektivität des Benfordschen Gesetzes in der Wirtschaftsprüfung. So zeigen beispielsweise Jann und Diekmann (2010) auf, dass die Anwendung der Benford Verteilung ein problematisches Tool zur Unterscheidung zwischen manipulierten und nicht manipulierten Schätzungen sei. Auch Bauer und Gross (2010) sowie Diekmann (2007) legen in ihren Werken Beweise vor, dass die Wirksamkeit von Benford Analysen, um Betrugsfälle aufzudecken, gering zu sein scheint.

3.1.2 Dichte-basierte Methoden

Ein weiteres Vorgehen, Ausreißer in einem Datenset zu erkennen, stellen die dichte-basierten Ansätze dar.

Diese folgen der generellen Idee, dass die Dichte eines Datenpunktes mit der Dichte um seine lokalen Nachbarn verglichen werden kann. Die Berechnung der Dichte eines Datenobjekts basiert auf nahen gelegenen Punkten, die in einem festgelegten Radius liegen. (Chepenko, 2018)

Die verschiedenen Techniken unterliegen laut Kriegel et al (2009) der Annahme, dass auf der einen Seite die Dichte, welche einen normalen Datenpunkt umgibt, ähnlich zu der Dichte seiner Nachbarn sei und auf der anderen Seite die Dichte um einen Ausreißer herum sich erheblich von der Dichte dessen Nachbarn unterscheiden.

Zusätzlich werden Outlier Scores, basierend auf der Berechnung der Dichte, errechnet. Dieser Score gibt neben der Klassifizierung, ob der Datenpunkt ein Ausreißer ist oder nicht, zusätzlich an, wie stark das jeweilige Datenobjekt vom normalen Verhalten abweicht (Kriegel, Kröger, Schubert, & Zimek, 2009).

Dieser Ansatz der Outlier Scores wurde zuerst von Breuning et al (2000) eingeführt und basiert auf dem mathematischen Konzept des Local Outlier Factor (LOF). Dieser ist definiert als der Durchschnitt der Verhältnisse der Dichte von Datenpunkt p und der Dichte seiner nächsten Nachbarn (Breunig, Kriegel, Ng, & Sander, 2000). Die Nachbarschaft wird beschrieben durch die Entfernung zu den „MinPts" nächsten Nachbarn, wobei „MinPts" die minimale Anzahl an nächsten Nachbarn bezeichnet (Hewahi & Saad, 2007).

Obwohl der dichte-basierte Ansatz und die Berechnung des LOF in vielen Domänen zur Erkennung von Ausreißern zum Einsatz kommt, stellen Agyemang & Ezeife (2004) in ihren Forschungen Limitationen des Algorithmus dar. So beschreiben sie, dass der Hauptnachteil in der Berechnung der Erreichbarkeitsentfernung der Nachbarn liegt. Die Autoren geben an, dass die Errechnung der Entfernungen bei großen „MinPts" sehr teuer werden kann. Darüber hinaus wird aufgezeigt, dass der LOF für jeden gegebenen Datenpunkt berechnet werden muss, bevor die vereinzelten Ausreißer identifiziert werden können. Dies sei laut Agyemang & Ezeife (2004) kein wünschenswertes Vorgehen, da Ausreißer nur einen kleinen Bruchteil des gesamten Datensatzes ausmachen.

3.1.3 Distanz-basierte Methoden

Distanz-basierte Ansätze wurden zunächst von Knorr und Ng (1998) vorgestellt. Ausreißer werden hier wie folgt definiert: „An Object O in a dataset T is a DB(p,D)-outlier if at least fraction p of the objects in T lies greater than distance D from O" (Knorr & Ng, 1998, S. 393)

Mit gegebenem Radius ε und Prozentsatz ß (z.B. 0,5%) stellen die Punkte p1 und p2 in folgender Abbildung einen distanz-basierten Ausreißer dar, da höchstens ß Prozent aller anderen Punkte des Datensatzes eine Distanz zu p1 und p2 aufweisen dürfen, die kleiner als ε ist. Demzufolge werden die Punkte p1 und p2 als Ausreißer klassifiziert.

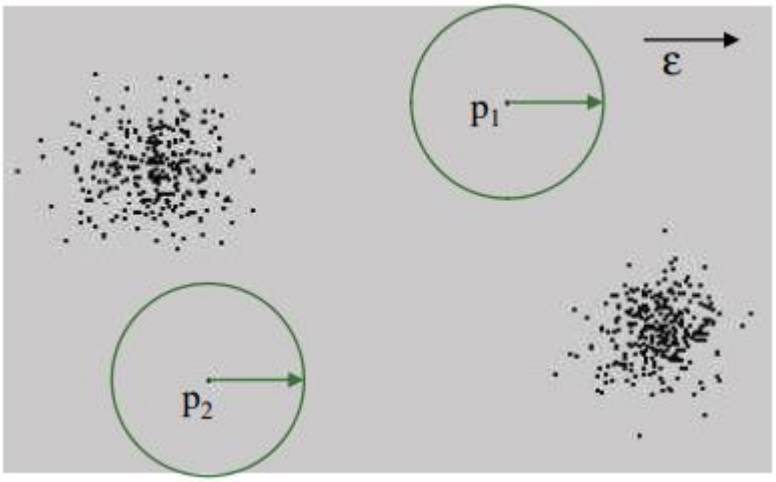


Abbildung 3: Distanz-basierte Ausreißer (Kriegel, Kröger, & Zimek, 2010, S. 33)

Ferner geben Knorr et al (1998) an, dass die Notation von distanz-basierten Ausreißern mit der Definition von Hawkins einhergeht und dadurch eine bessere Rechenkomplexität aufzuweisen ist.

Das Konzept wird durch Ramaswamy et al (2000) erweitert. Hierbei werden die Datenobjekte basierend auf ihrer Distanz zu seinen k nächsten Nachbarn eingestuft. Zusätzlich werden die obersten k Punkte als Ausreißer markiert. Dieser Ansatz der k nächsten Nachbarn (kNN) hat die Vorteile, dass die Datenpunkte danach geordnet werden können, wie stark sie sich von den normalen Objekten unterscheiden. Weiterhin ist der Algorithmus nicht abhängig von einem fixen Distanz Parameter D.

Dennoch dokumentiert Petrovskiy (2003) einige Nachteile der distanz-basierten Ansätze. Zunächst besitzen viele Datensätze in modernen IT-Systemen heterogene Datenstrukturen. Diese unterschiedlichen Strukturen erschweren die Berechnung der Distanz zwischen Datenobjekten erheblich. Zusätzlich hebt der Autor hervor, dass die vorhandenen Algorithmen abhängig von vorher definierten Parametern seien, zum Beispiel die Anzahl der Nachbarn k oder die Distanz D. Verändern sich diese, muss das komplette Model neu berechnet und konstruiert werden.

3.1.4 Isolations-basierte Methoden

Diese Strategie verfolgt einen differenzierteren Ansatz als die bisher vorgestellten Konzepte. So werden laut Domingues et al (2017) einzelne Instanzen des Datensets isoliert, ohne Dichte- oder Entfernungsmessungen durchzuführen. Ferner geben sie an, dass die grundlegende Idee darin besteht, Ausreißer vom Rest der Datenpunkte zu trennen, anstatt „normale“ Punkte zu identifizieren.

Das Konzept von Isolation Forest beziehungsweise iForest wurde von Liu et al (2008) beschrieben. Hierbei wird eine Baumstruktur konstruiert, um einzelne Instanzen

gezielt zu isolieren. Diese Struktur wird durch wiederholtes, rekursives Trennen der Instanzen basierend auf deren Attributwerten aufgebaut, bis alle Punkte des Datensets in einzelnen Knoten isoliert sind. Daraufhin wird der Score eines Datenobjekts berechnet, indem die durchschnittliche Pfadlänge von der Wurzel des Baums bis zu dem Knoten, der den Datenpunkt beinhaltet, zur Berechnung herangezogen wird. Hierbei deutet ein kurzer Pfad auf ein Datenobjekt hin, das sich deutlich von den normalen Punkten unterscheidet, da seine Attributwerte schnell und leicht zu isolieren sind.

3.1.5 Cluster-basierte Methoden

Cluster-basierte Methoden, oder auch Clustering genannt, beschreiben den Prozess, Objekte eines Datensets in vorher unbekannte, aber konzeptionell zusammenhängende Gruppen beziehungsweise Cluster aufzuteilen. Datenpunkte in einem Cluster sind sowohl ähnlich zueinander als auch unähnlich zu Objekten der anderen Cluster. (Han, Kamber, & Pei, 2012)

He et al (2003) stellen eine neue Definition für Ausreißer im Kontext des Clustering vor. Sie beschreiben, dass Ausreißer aus der Sicht der Cluster identifiziert werden können, indem alle Datenpunkte, die nicht in großen Clustern liegen, als Ausreißer deklariert werden. Darüber hinaus stellen sie zwei Formeln vor, damit die Unterteilung in große und kleine Cluster und somit die Identifizierung von Ausreißern durchgeführt werden kann.

Ein in vielen Anwendungen und Domänen vertretener cluster-basierter Algorithmus, ist der sogenannte K-means Algorithmus oder K-means Clustering. Dabei existieren viele unterschiedliche Ausprägungen und Varianten des Algorithmus. Die generelle Idee besteht darin, für jeden Datenpunkt einen Zugehörigkeitsgrad zu berechnen, um festzulegen in welchem Cluster der Punkt zugeordnet werden kann. K steht hierbei für die Anzahl der vorher festgelegten Cluster. (Kanungo, Netanyahu, & Wu, 2002) Diese Berechnung wird vermehrt mit Hilfe der Euklidischen Distanz durchgeführt. Außerdem werden diese Distanzen mit Hilfe einer Error-Funktion minimiert, um die Berechnung der Cluster abzuschließen. (Jain, 2010) Diese ist in folgender Grafik dargestellt.

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} |i_l - w_j|^2$$

$$j \in \{1, \ldots, k\}, l \in \{1, \ldots, n\}$$

Abbildung 4: Error-Funktion (Anton, Kanoor, Fraunholz, & Schotten, 2018, S. 7)

Dennoch weisen Singh et al (2011) darauf hin, dass der K-means Algorithmus Limitationen unterliegt. Sie führen aus, dass der Wert K a priori als Parameter für den Algorithmus angegeben werden muss. Deshalb kann es sich als schwierig darstellen, die Anzahl der Cluster vorher zu bestimmen. Dies kann bei zweidimensionalen Daten zwar durch visuelle Inspektion erfolgreich durchgeführt werden, stellt den Anwender aber bei Daten von höheren Dimensionen durchaus vor Probleme.

3.1.6 Support Vektor-basierte Methoden

Ein weiterer Ansatz zur Erkennung von Anomalien besteht darin, ein Modell der Daten zu konstruieren, das eine beschreibende Entscheidungsgrenze aufweist. Hierbei unterliegen die angewandten Methoden der Annahme, dass die überwiegende Mehrheit der Datenpunkte normal ist. (Emmott, Das, Dietterich, Fern, & Wong, 2016)

Support Vector Maschinen (SVM) wurden zunächst von Boser et al (1992) vorgestellt. Die grundlegende Idee besteht darin, eine Abgrenzung zwischen zwei Gruppen zu erstellen, damit jede einzelne Dateninstanz den größtmöglichen Abstand zur Abgrenzung hat.

Ein bekannter Algorithmus zur Aufdeckung von Ausreißern über Support Vector-basierte Methoden, der One-Class SVM Algorithmus, wurde durch Schölkopf et al (1999) eingeführt. Hierbei wird ein Großteil der Daten durch eine große Marge vom Ursprung getrennt. Eine geometrische Interpretation des Algorithmus ist in Abbildung 5 dargestellt. Zusätzlich zeigen Li et al (2003) auf, dass alle Datenpunkte, die nahe am Ursprung liegen, als Ausreißer oder anomale Datenpunkte betrachtet werden können. Somit kann für diese Punkte ein entsprechender Outlier Score berechnet werden.

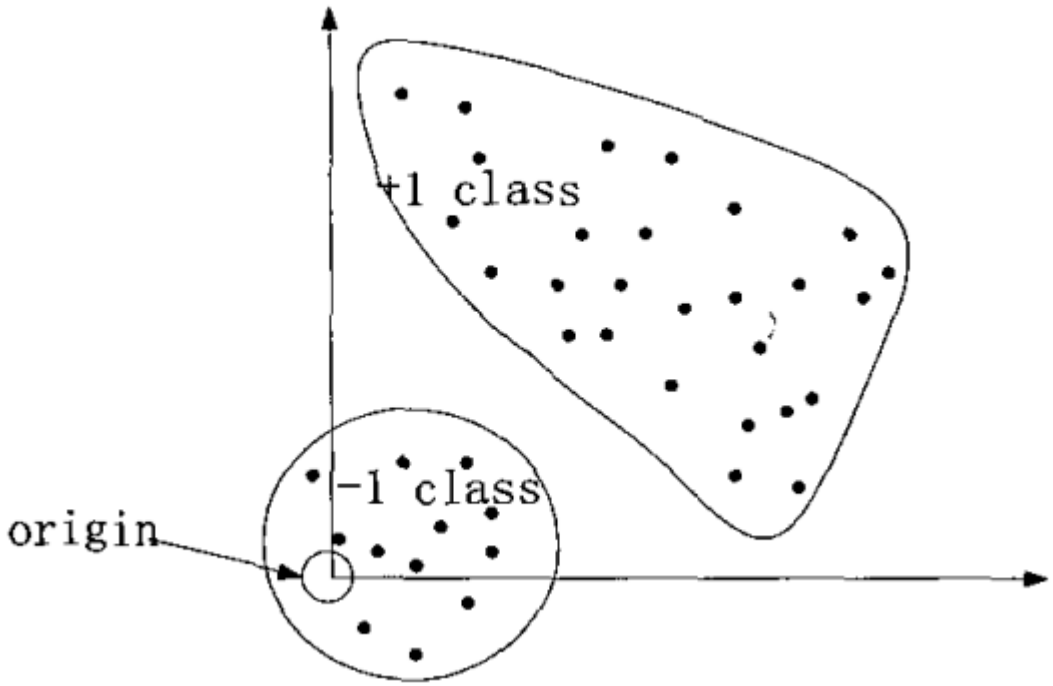


Abbildung 5: Geometrische Interpretation des One-Class SVM (Li, Huang, Tian, & Xu, 2003, S. 3078)

Allerdings dokumentieren Huang & LeCun (2006), dass diese Ansätze unter bestimmten Voraussetzungen Limitationen unterliegen können. Zum einen sei die Verarbeitung der Daten sehr rechenintensiv und skaliert schlecht mit der Größe der Datensets. Zum anderen beschreiben die Autoren, dass der Einsatz von SVM-basierten Methoden bei der Anwendung von hochdimensionalen Datensets ab 10^4 Attributen sowie sehr großen Daten ab 10^5 bis 10^6 Proben limitiert ist.

3.2 Methoden zur Evaluierung von Ausreißeranalysen

Laut Anton et al (2018) kann die Anwendung von Ausreißeranalysen als binäres Klassifikationsproblem betrachtet werden. Hierbei bildet jeder Datenpunkt entweder eine normale oder anomale Instanz ab. In den folgenden Unterkapiteln werden Möglichkeiten vorgestellt, Machine Learning Modelle zu evaluieren und deren Leistung miteinander zu vergleichen.

3.2.1 Klassische Metriken

Zur Evaluierung von Klassifikationsproblemen können nach Han et al (2012) verschiedene Metriken zur Hand gezogen werden. Hierfür müssen zunächst folgende Kennzahlen definiert werden:

- True Positives (TP): Anzahl der Objekte, die als Anomalie gekennzeichnet und vom Algorithmus erkannt wurden.

- True Negatives (TN): Anzahl der Objekte, welcher der Algorithmus als Ausreißer klassifiziert hat, obwohl diese gar keine sind.
- False Positives (FP): Anzahl der Objekte, die als Anomalien gekennzeichnet, aber vom Algorithmus nicht erkannt wurden.
- False Negatives (FN): Anzahl der Objekte, die als normal gekennzeichnet und vom Algorithmus ebenfalls als normal eingestuft wurden.

Diese Kennzahlen lassen sich in der sogenannten Konfusion-Matrix darstellen, welche in folgender Abbildung zu sehen ist.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Abbildung 6: Konfusion-Matrix (AIMultiple, 2020, S. 1)

Die Maße precision und recall werden in vielen Anwendungen und Domänen zur Evaluierung herangezogen. Hierbei kann precision als Maß der Exaktheit verstanden werden. Es gibt die Prozentzahl der Objekte an, die vom Algorithmus als positiv identifiziert wurden und auch tatsächlich positiv sind. Wohingegen recall als Maß der Vollständigkeit gesehen werden kann. Dieses gibt an, welche Prozentzahl der positiven Datenpunkte vom Algorithmus erkannt wurde. (Han, Kamber, & Pei, 2012) Die vorgestellten Maße berechnen sich folgendermaßen:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P}.$$

Abbildung 7: Berechnung der Maße precision und recall (Han, Kamber, & Pei, 2012, S. 368)

Dennoch unterliegen die beschriebenen Metriken einer entscheidenden Limitation. Um die Leistung verschiedener Algorithmen mittels der Maße zu vergleichen, werden

Labels im Datenset zur Berechnung der Kennzahlen benötigt (Goldstein & Uchida, 2016). Die zugrundliegenden Algorithmen müssen daher aus dem Bereich des überwachten Lernens stammen.

3.2.2 Receiver operating characteristics

Der receiver operating characteristics (ROC) Graph stellt laut Fawcett (2006) eine weitere Methode zur Evaluierung von binären Klassifikationsproblemen dar. Hierbei wird die Performance der Algorithmen mittels Visualisierung abgebildet. Der ROC Graph wird als zweidimensionaler Graph dargestellt, in welchem die TP Rate auf der Y-Achse und die FP Rate auf der X-Achse eingezeichnet werden. Die FP Rate wird durch das Verhältnis von False Positives zur Anzahl aller negativen Datenpunkte errechnet. Die TP Rate wohingegen repräsentiert den bereits beschriebenen recall (Fawcett, 2003). In folgender Abbildung ist eine beispielhafte Darstellung einer ROC Kurve zu sehen.

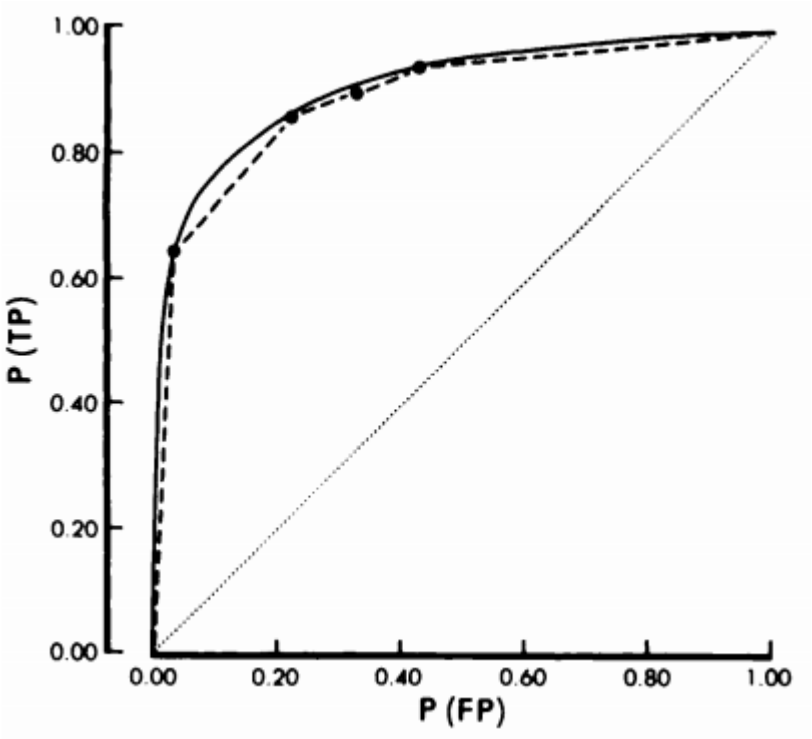


Abbildung 8: ROC Kurve (Hanley & McNeil, 1982, S. 33)

Darüber hinaus legt Chan dar, dass „je näher die ROC Kurve der 45 Grad Diagonale des ROC-Raums kommt, desto ungenauer ist der Test“ (Chan, 2018, S. 1). Nichtsdestotrotz zeigen Lasko et al (2005) auf, dass ROC Kurven bei der Bewertung eines Tests nützlich sind. Dennoch tritt oftmals ein Verlangen nach einem einzelnen Index auftritt, um die Performance und Genauigkeit eines Tests zu messen. Der am meisten verwendete Index, ist laut Hanley et al (1982) die Fläche unter der Kurve (engl.: AUC, Area under

the curve). Der Wert wird mit Hilfe des Integrals unter der gesamten Kurve von (0,0) bis (1,1) berechnet (Google LLC, 2020).

Je höher der AUC ist, desto besser klassifiziert der Algorithmus und kann zwischen Ausreißern und normalen Datenpunkten unterscheiden. Im Abbildung 9 ist eine optimale ROC Kurve abgebildet. Der Flächenwert eins unter der Kurve zeigt, dass der angewandte Algorithmus exakt zwischen Anomalien und gewöhnlichen Datenobjekten differenzieren kann (Narkhede, 2018).

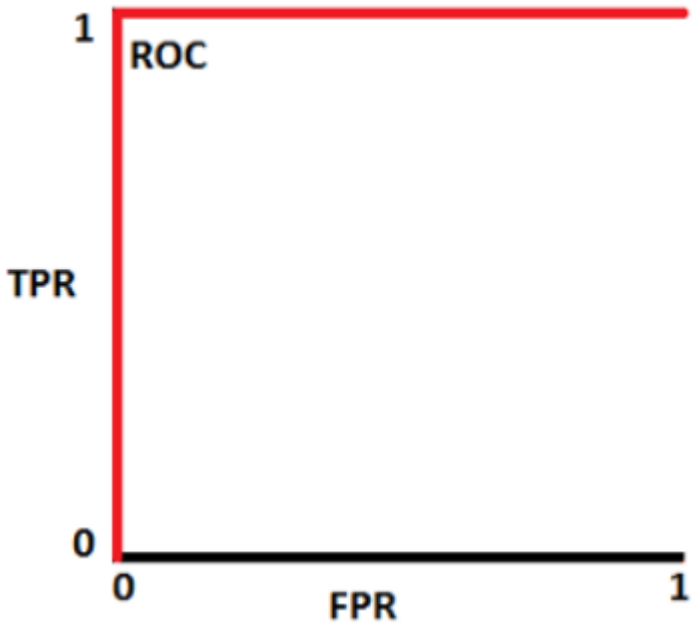


Abbildung 9: ROC Kurve mit AUC = 1 (Narkhede, 2018, S. 1)

Jedoch zeigen Han et al (2012) auf, dass für die zwei vorgestellten Evaluierungsmöglichkeiten ein Labeling der Daten benötigt wird, um die Methoden erfolgreich anwenden zu können. Da in der realen Welt und bestimmten Anwendungsgebieten ungelabelte Datensets vorkommen, bedarf es zusätzlicher Evaluierungstechniken.

3.2.3 Domänen Experte

Eine Alternative zur Evaluierung von Algorithmen, basierend auf den klassischen Metriken, stellt die visuelle Be- und Auswertung der Ergebnisse mit Hilfe eines Domänen Spezialisten dar.

Hierbei werden nach Lui et al (2017) die Analyseergebnisse von Sachkundigen des entsprechenden Anwendungsbereiches überprüft und mit Hilfe des Domänenwissen der Experten miteinander verglichen. Ferner kann eine visuelle Aufarbeitung der Resultate den Experten unterstützen. Allerdings beschreiben Achtert et al (2010), dass sich die Interpretation der Ergebnisse und daher eine Schlussfolgerung als schwierig

herausstellt, da die bereitgestellten Outlier Scores der verschiedenen Algorithmen in der Skalierung unterschiedlich sind. Des Weiteren zeigen sie auf, dass für die den betrachteten Anwendungsfall oftmals keine allgemeingültige Definition eines Ausreißers vorliegt.

3.2.4 Excess Mass- & Mass Volume Kurve

Eine weitere Möglichkeit zur Bewertung der Analyseergebnisse stellen die von Goix (2016) eingeführten Metriken dar, welche auf den Konzepten der Excess Mass (EM)- und Mass-Volume (MV) Kurve basieren. Der Autor zeigt auf, dass in vielen Situationen wenige oder keine Labels in den zu analysierenden Datensets vorliegen. Somit sei der Bedarf an Evaluierungsmetriken, welche die Performance von unüberwachten Algorithmen miteinander vergleichen können, sehr hoch.

Goix et al (2015) legen darüber hinaus dar, dass die grundlegende Idee hinter der EM-Kurve darin besteht, eine Lagrange Formulierung eines eingeschränkten Minimalisierungsproblems zu betrachten. In folgender Abbildung sind verschiedene EM-Kurven basierend auf der Verteilung der Datenpunkte zu sehen. Eine flache EM-Kurve entsteht beispielsweise durch eine Normalverteilung der Daten oder eine Verteilung mit schweren Rändern.

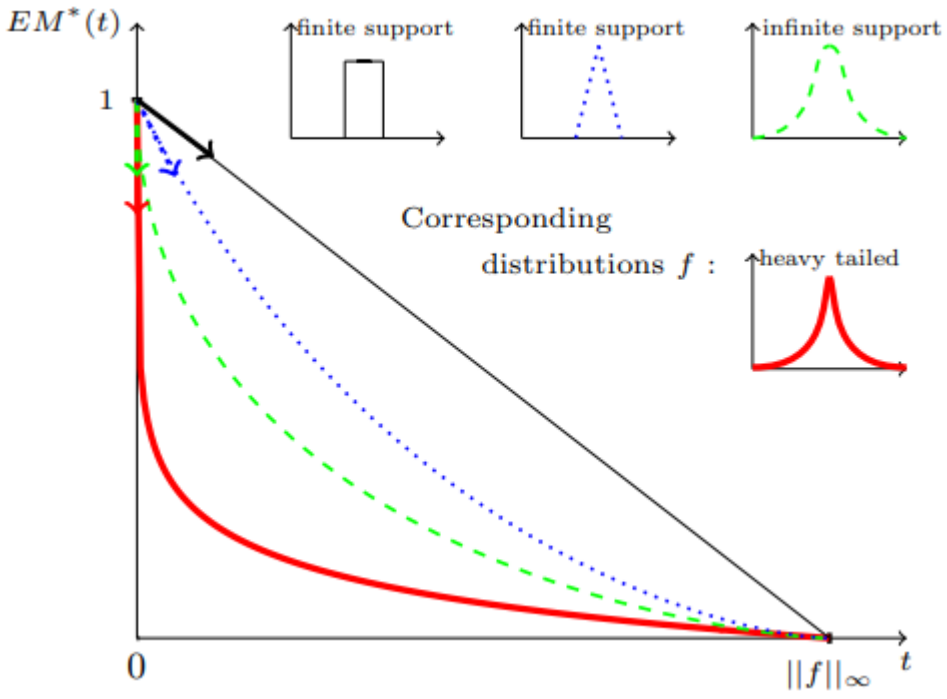


Abbildung 10: EM-Kurven in Abhängigkeit von der Verteilung der Datenpunkte (Goix, Sabourin, & Clémencon, 2015, S. 3)

Zur Berechnung der Metriken sowie Anzeigen der Kurven stellt eine scoring Funktion S die Voraussetzung des EM- und MV Kurven Ansatzes dar. Ähnlich dem bereits

vorgestellten Outlier Score, definiert eine scoring Funktion S ein Ranking der observierten Datenpunkte. Sie gibt eine Reihenfolge der Objekte vor, welche als Grad der Abnormalität interpretiert werden können. Desto niedriger der Wert S(x) desto abnormaler sei der Datenpunkt x. (Goix, 2016) Die EM- und MV Kurven können mathematisch folgendermaßen abgebildet werden:

$$MV_s(\alpha) = \inf_{u \geq 0} \ \mathrm{Leb}(s \geq u) \ \ s.t. \ \ \mathbb{P}(s(\mathbf{X}) \geq u) \geq \alpha$$
$$EM_s(t) = \sup_{u \geq 0} \ \mathbb{P}(s(\mathbf{X}) \geq u) \ - \ t\mathrm{Leb}(s \geq u)$$

Abbildung 11: Mathematische Formel der EM - und MV Kurve (Goix, 2016, S. 2)

Die MV-Kurven Metrik wurde von Clémencon et al (2013) vorgestellt und basiert auf dem Konzept der minimum volume sets. Die MV-Kurve repräsentiert eine partielle Quasiordnung der gegebenen scoring Funktionen. Außerdem wird die Sammlung von optimalen Elementen definiert als: „the set of scoring functions whose MV-curve is minimum everywhere“ (Clémençon & Jakubowicz, 2013, S. 2). Zudem kann die MV-Kurve als Erweiterung der ROC-Kurve in einer unüberwachten Umgebung gesehen werden (Clémencon & Thomas, 2017).

Des Weiteren stellen Goix et al (2016) einen Vergleich der Performance von Metriken basierend auf gelabelten Daten, im genaueren ROC und Precision-Recall Kurven und den Konzepten der MV und EM-Kurve auf. Dabei wurden die Leistung von drei AD Algorithmen (One-Class SVM, Isolation Forest und Local Outlier Factor) auf 12 verschiedenen Datensets untersucht. Die Autoren kommen zu dem Ergebnis, dass die Metriken für unüberwachtes Lernen in ungefähr 80 Prozent der Fälle in der Lage sind, den besten Algorithmus für den Anwendungsfall zu identifizieren. Darüber hinaus legt der Autor dar, dass die optimalen EM- und MV-Kurven mathematisch folgendermaßen beschrieben werden können. Hierbei stellt zum einen der niedrigste Mass Volume Wert, zum anderen der höchste EM-Wert das optimale Ergebnis dar.

$$MV^*(\alpha) \leq MV_s(\alpha) \text{ for all } \ \alpha \in (0,1)$$
$$EM^*(t) \geq EM_s(t) \text{ for all } t > 0.$$

Abbildung 12: Optimale MV- und EM-Werte (Goix, 2016, S. 2)

4. Implementierung

Nachdem im vorangegangenen Kapitel verschiedene Ansätze und Konzepte zur Ausreißererkennung und Evaluierung der Ergebnisse identifiziert wurden, soll im folgenden Abschnitt mit Hilfe der Anwendung von drei ausgewählten AD Algorithmen Teilfrage 3 beantwortet werden:

- **TF3)** Wie können Machine Learning Algorithmen zur Betrugserkennung in der Abschlussprüfung angewandt werden?

Hierfür werden zunächst die zur Verfügung stehenden Datensets sowie der in der Arbeit betrachtete Anwendungsfall beschrieben. Anschließend werden die in der Literaturanalyse vorgestellten Algorithmen und Metriken zur Evaluierung anhand verschiedener Kriterien miteinander verglichen und dementsprechend für die Anwendung im untersuchten Szenario ausgewählt. Zusätzlich werden die zur Aufbereitung der Datensets benötigten Preprocessing Schritte erläutert und abschließend die Implementierung der empirischen Untersuchung vorgestellt.

4.1 Beschreibung der Daten

Zur Beantwortung der Forschungsfrage stehen zwei Datensets zur Verfügung. Diese künstlich erstellten Daten sind an reelle Werte von Unternehmen angelehnt und bilden typische Buchungsdatensätze zwei verschiedener Branchen ab. Für die Datensets stehen 323906 (Datenset A) sowie 68242 (Datenset B) Datenobjekte zur Verfügung.

Die beiden zu analysierenden Datensets weisen folgende Struktur auf:

Spalte	Datentyp	Beschreibung
Ktonr	Alphanumerisch	Kontonummer
KtoBez	Alphanumerisch	Kontobezeichnung
Buchungstext	Alphanumerisch	Buchungstext
Belegdatum	Datum	Belegdatum
BelegNr	Alphanumerisch	Belegnummer
Gktonr	Alphanumerisch	Gegenkontonummer
Umsatz_Soll	Alphanumerisch	Höhe der Buchung auf dem Sollkonto
Umsatz_Haben	Alphanumerisch	Höhe der Buchung auf dem Habenkonto
BuSchl	Numerisch	Buchungsschlüssel
UStSatz	Numerisch	Umsatzsteuer Satz
Buchungsdatum	Datum	Buchungsdatum

UStNr	Alphanumerisch	Umsatzsteuer Nummer
UStLand	Alphanumerisch	Umsatzsteuer Land
Betrag	Numerisch	Betrag der Buchung
SHKZ	Alphanumerisch	Soll / Haben Kennzeichen
BuchID	Numerisch	ID der Buchung
Benutzer	Alphanumerisch	ID des Anwenders im Buchungssystem
Währung	Alphanumerisch	Währung der Buchung
Belegwährung	Alphanumerisch	Währung des Belegs

Tabelle 2: Beschreibung der Daten

4.2 Beschreibung des Anwendungsfalls

Im Prozess der digitalen Datenanalyse im Rahmen der Abschlussprüfung stellt das sogenannte Journal Entry Testing (JET) einen Grundbaustein der risikoorientierten Prüfung dar (EBS audit & accounting, 2020). Sowohl die internationalen Prüfungsstandards 240: „Die Verantwortung des Abschlussprüfers bei dolosen Handlungen“ als auch die IDW Verlautbarungen 210: „Zur Aufdeckung von Unregelmäßigkeiten im Rahmen der Abschlussprüfung“ empfehlen zur ordnungsgemäßen Durchführung von Jahresabschlussprüfungen den Einsatz von JET Analysen (DATEV eG, 2020).

Der im weiteren Verlauf der Arbeit zu analysierende Anwendungsfall ergibt sich aus der Kombination zweier JET Analysen:

1. Anzahl Buchungen pro Erfasser
2. Buchungen bei Umsatzerlöskonten

Dadurch soll einerseits die Analyse auf einen der am häufigsten bebuchten Kontenbereiche konzentriert werden, andererseits Zusammenhänge zwischen ungewöhnlichen Buchungen und den Anwendern des Buchhaltungssystems festgestellt werden. Die Algorithmen erhalten als Input den errechneten Saldo beziehungsweise die Höhe der Buchung, die dem Kontenbereich der Umsatzerlöse zugeordnet werden kann, sowie den jeweiligen Nutzer.

4.3 Auswahl der Algorithmen und Evaluierungstechniken

Aus der Struktur und den Attributen der Datensätze ergibt sich, dass diese keine Labels aufweisen. Daher müssen die ausgewählten Algorithmen auf unüberwachtem maschinellen Lernen basieren. Es besteht zwar die Möglichkeit, die Datenpunkte von mehreren, unabhängigen Experten untersuchen und einstufen zu lassen und somit

eigene Labels zu kreieren. Allerdings stellt sich dies in der Domäne der Abschlussprüfung als sehr komplex und zeitaufwendig heraus. Im Rahmen dieser Bachelorarbeit wurde deshalb darauf verzichtet.

Des Weiteren soll eine Klassifikation, ob ein bestimmter Datenpunkt als Ausreißer eingestuft wird oder nicht als Ergebnis des Algorithmus angezeigt werden. Zusätzlich dazu soll der Algorithmus eine Gewichtung wie stark dieser Punkt ausreißt beziehungsweise sich von den normalen Punkten unterscheidet angeben. Im Zuge von Kapitel 3 wurden bereits verschiedene Ansätze beziehungsweise Methoden zur Erkennung von Ausreißern inklusive eines dazugehörigen Algorithmus beschrieben.

In folgender Matrix ist dargestellt, welche Voraussetzungen, bedingt durch den Anwendungsfall und den zur Verfügung stehenden Datensatz, die vorgestellten Algorithmen erfüllen. Hierbei gelten drei Voraussetzungen:

- Unüberwacht: Der Algorithmus muss ohne die Hilfe von vorher bekannten Zielwerten beziehungsweise Labels aus den Eingabedaten Ausreißer erkennen.
- Output Klassifikation: Der Algorithmus muss zwischen normalen und anormalen Datenpunkten unterscheiden und die Datenobjekte in zwei Klassen einteilen können.
- Output Outlier Score: Der Algorithmus muss die untersuchten Datenpunkte danach gewichten, wie stark sich diese vom normalen Verhalten unterscheiden.

Voraussetzungen / Algorithmus	Unüberwacht	Output Klassifikation	Output Outlier Score
Local Outlier Factor	X	X	X
K Nächste Nachbarn			
Isolation Forest	X	X	X
K-Means	X	X	
One-Class SVM	X	X	X

Tabelle 3: Auswahlmatrix für vorgestellte Algorithmen

Aus der Tabelle ergibt sich, dass der K Nächste Nachbarn Algorithmus keine der Bedingungen erfüllt und daher nicht verwendet werden kann. Die Fähigkeit einen Outlier Score anzugeben, fehlt beim Algorithmus K-Means, weshalb dieser ebenfalls nicht angewendet werden kann. Drei der vorgestellten Algorithmen, der Local Outlier Factor, Isolation Forest und One-Class SVM erfüllen alle beschriebenen

Voraussetzungen für den Anwendungszweck und sollen als Grundlage für die Analyse dienen.

Zur Überprüfung der Ergebnisse, stehen aufgrund der ungelabelten Datensätze lediglich zwei der vorgestellten Methoden zur Evaluierung zur Verfügung. Sowohl die klassischen Metriken als auch der ROC Graph benötigen gelabelte Daten und können folglich im gegebenen Anwendungsfall nicht zum Einsatz kommen.

Dennoch können entweder Domänen Experten zur manuellen Überprüfung der Ergebnisse hinzugezogen, oder die von Goix vorgestellten Konzepte der Excess Mass- & Mass Volume Kurve angewendet werden. Allerdings stellt sich, wie bereits in Kapitel 3 beschrieben, eine manuelle Evaluierung, vor allem wegen des großen Interpretationsspielraums als schwierig heraus.

Aus diesem Grund und da Goix (2016) bereits dargelegt hat, dass die Metriken EM- und MV-Kurve eine Performance ähnlich der ROC- und Precision-Recall Kurven aufweisen, werden im Rahmen dieser Arbeit die EM- und MV-Kurven zur Evaluierung der Analyseergebnisse eingesetzt.

4.4 Datenaufbereitung (Preprocessing)

Um sowohl qualitativ hochwertige Ergebnisse zu erhalten als auch den gegebenen Datensatz auf den Anwendungsfall zuzuschneiden, werden die Daten mit Hilfe folgender Schritte aufbereitet:

1. **Filtern der Spalten**: Für das betrachtete Szenario werden die essenziellen Spalten Kontonummer, ‚Umsatz Soll', ‚Umsatz Haben' und ‚Benutzer' aus dem Datensatz extrahiert.
2. **Fehlende Werte in den Spalten werden ersetzt**: Für die Spalten ‚Umsatz Soll' und ‚Umsatz Haben' werden die fehlenden Werte durch den Wert 0 ersetzt. Für ‚Kontonummer' werden die Datenpunkte gelöscht. Bei der Spalte ‚Benutzer' wird eine neue Kategorie „Unbekannt" eingeführt und alle fehlenden Werte zu dieser zugeordnet.
3. **Filtern der Spalte ‚Kontonummer'**: Für den Kontenbereich der Umsatzerlöse werden nur bestimmte Kontonummern benötigt.
4. **Datentyptransformation**: Die Spalte ‚Benutzer' wird zur kategorischen Variablen umgewandelt.
5. **Berechnung der Höhe der Umsatzerlöse**: Basierend auf den Spalten ‚Umsatz Soll' und ‚Umsatz Haben', wird der Saldo für alle einzelne Buchung berechnet.
6. **Entfernen der nicht mehr benötigter Spalten**: ‚Umsatz Soll', ‚Umsatz Haben' und ‚Kontonummer' werden zum Trainieren des Algorithmus entfernt.

Dieser Prozess der Datenvorbereitung muss für jeden gegebenen Datensatz durchgeführt werden und stellt für alle der drei Algorithmen den ersten Schritt im Bearbeitungsprozess dar. Die verarbeitenden Daten sollen anschließend als Input für die ausgewählten Algorithmen dienen.

In Grafik 13 ist das Ergebnis des Datenvorbereitungsprozesses für Datensatz B als Plot dargestellt. Auf der X-Achse ist die ID des entsprechenden Sachbearbeiters abgebildet, wohingegen auf der Y-Achse die Höhe der entsprechenden Buchung angezeigt wird.

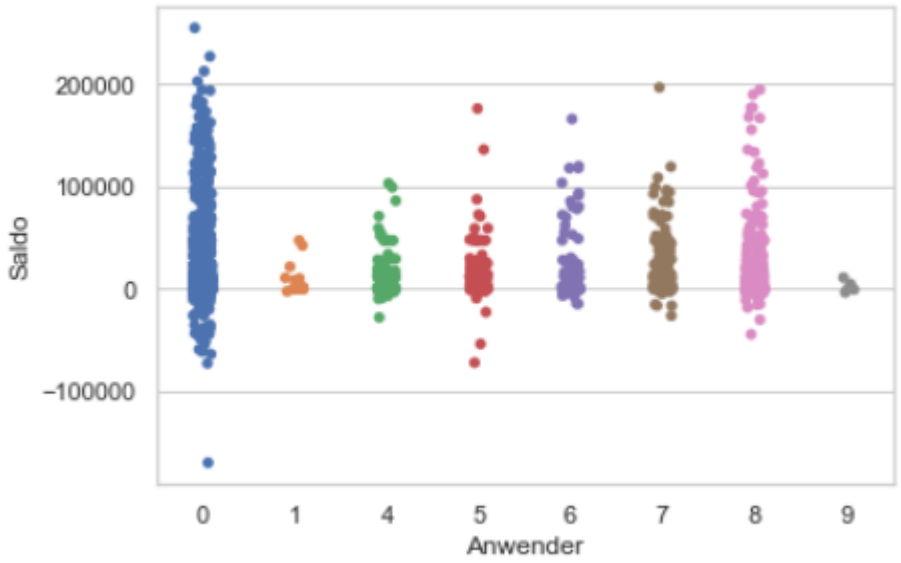


Abbildung 13: Datensatz B nach vollständigem Preprocessing

4.5 Implementierung

Die Implementierung erfolgt in drei Python Modulen und stellt für jeden der drei Algorithmen sowie den zwei Datensets den gleichen Prozess dar:

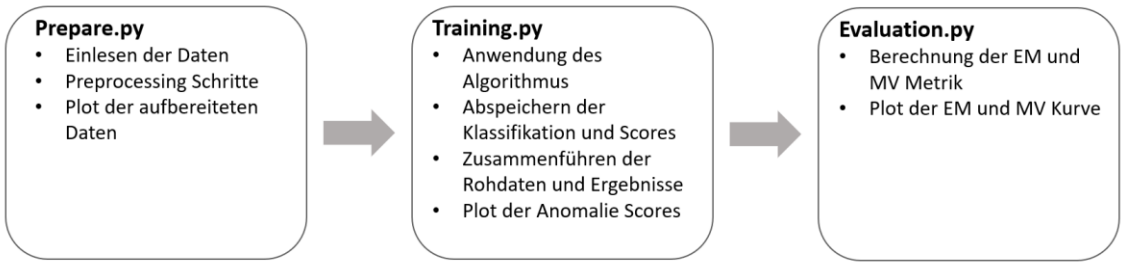


Abbildung 14: Architektur der Anwendung

Prepare.py

Innerhalb der ersten Phase werden vorbereitende Maßnahmen getroffen und die Rohdaten werden mit Hilfe des Preprocessing in eine geeignete Form für die Anwendung des jeweiligen Algorithmus überführt.

Der folgende Abschnitt zeigt das Einlesen und Preprocessing der Daten:

```
1  import pandas as pd
2
3  df = pd.read_csv(file_path ,sep=';',header=None, encoding='ISO-8859-1')
4  do_preprocessing(df)
```

Abbildung 15: Einlesen der Daten

<prepare.py > [1-4]

Zunächst liest das Modul mit Hilfe der Methode „read_csv“ ein Dataframe-Objekt „df“ ein, das als Datenstruktur für die CSV-basierten Inhalte dienen soll. Hierbei gibt die Variable „file_path“ den Pfad zum gegebenen Datensatz an. Der Parameter „sep“ bestimmt das zu verwendende Trennzeichen, um die Daten korrekt einlesen zu können. Außerdem wird durch „header=None“ angegeben, dass im Datensatz keine Spaltennamen in der ersten Zeile eingetragen sind. Zusätzlich wird mittels „encoding“ festgelegt, welche Codierung beim Einlesen verwendet werden soll. (pandas, 2020) Anschließend erfolgt das Preprocessing der Daten über den Aufruf der Methode „do_preprocessing“. In folgender Abbildung ist der erste Teil dieser Funktion abgebildet.

```
7  def do_preprocessing(df):
8
9      # Filtern der Spalten (Schritt 1)
10     df= df[['KtoNr','Umsatz_S', 'Umsatz_H', 'Anwender']]
11
12     # Handle missing values (Schritt 2)
13     df['KtoNr'].dropna(inplace=True)
14     df['Umsatz_S'].fillna(0, inplace=True)
15     df['Umsatz_H'].fillna(0, inplace=True)
16     df['Anwender'].fillna('Unbekannt', inplace=True)
17
18     # Filtern der Kontonummern für den Kontenbereich Umsatzerlöse (Schritt 3)
19     df = df[((df['KtoNr'] >= 8000) & (df['KtoNr'] <= 8589)) | ((df['KtoNr'] >= 8900)
20         & (df['KtoNr'] <= 8919)) | ((df['KtoNr'] >= 8940) & (df['KtoNr'] <= 8959))
21         | ((df['KtoNr'] >= 8700) & (df['KtoNr'] <= 8799))].copy()
22
```

Abbildung 16: Methode do_preprocessing Schritt 1-3 <prepare.py> [7-22]

Anfänglich erfolgt das Filtern der Spalten im Dataframe, indem alle, nicht für den Anwendungsfall zu berücksichtigten Spalten, entfernt werden und ausschließlich mit den Spalten Kontonummer, ‚Umsatz Soll‘, ‚Umsatz Haben‘ und ‚Anwender‘ weitergearbeitet wird. Daraufhin wird Schritt 2 des Preprocessing, das Behandeln von fehlenden Werten ausgeführt. Hierfür werden alle Datenpunkte der Spalte Kontonummer ohne Wert mittels der Methode „dropna“ aus dem Datensatz entfernt. Die Daten der restlichen Spalten werden mit Hilfe der Methode „fillna“ aufgefüllt.

In den Spalten ‚Umsatz Soll' und ‚Umsatz Haben' wird der Wert 0 und in der Spalte ‚Anwender' bei fehlenden Objekten die neue Kategorie „Unbekannt" eingesetzt. Im Anschluss daran wird das Filtern der Kontonummern durchgeführt. Dazu werden ausschließlich die Datenpunkte zur weiteren Verarbeitung in Betracht gezogen, die dem Kontenbereich der Umsatzerlöse zugeordnet sind. Die weiteren Schritte der Verarbeitung sind in der folgenden Abbildung dargestellt.

```
23      # Datentyptransformationen (Schritt 4)
24      df['Umsatz_H'] = df['Umsatz_H'].str.replace(',','.')
25      df['Umsatz_H'] = pd.to_numeric(df['Umsatz_H'])
26
27      df['Umsatz_S'] = df['Umsatz_S'].str.replace(',','.')
28      df['Umsatz_S'] = pd.to_numeric(df['Umsatz_S'])
29
30      df['Anwender'] = df.Anwender.astype('category')
31      df['Anwender'] = df['Anwender'].cat.codes
32
33      # Berechnung des Saldos (Schritt 5)
34      df['Saldo'] = df.apply(lambda row: row.Umsatz_H - row.Umsatz_S, axis = 1)
35
36      # Entfernen nicht mehr benötigter Spalten (Schritt 6)
37      df = df.drop(columns=['Umsatz_S','Umsatz_H', 'KtoNr'])
38
39
40      return df
```

Abbildung 17: Methode do_preprocessing Schritt 4-6 <prepare.py> [23-40]

Der vierte Schritt des Preprocessing beinhaltet die Durchführung der Datentyptransformationen. Dafür wird zum einen die alphanumerische Variable ‚Anwender' in eine kategorische Variable umgewandelt, zum anderen die Spalten ‚Umsatz Haben' und ‚Umsatz Soll' in numerische Spalten transformiert. Weiterhin erfolgt in Schritt 5 die Berechnung des Saldos, indem für jede Datenreihe die Spalte ‚Umsatz Soll' von ‚Umsatz Haben' abgezogen wird. Dies geschieht mit Hilfe der Funktion „apply" und einer lambda Expression, welche die entsprechende Berechnung auf jede Zeile des Dataframes ausführt. Abschließend werden nicht mehr benötigte Spalten (‚Umsatz Soll', ‚Umsatz Haben' und ‚Kontonummer') in Schritt 6 entfernt.

Training.py

Die Implementierung der Algorithmen wurde im Rahmen dieser Arbeit nicht erneut durchgeführt, sondern es wird ein öffentliches Python Paket der Bibliothek ‚scikit-learn' verwendet, welches die drei zu untersuchenden Algorithmen bereits implementiert hat.

Zunächst empfängt das Modul Training.py die zuvor aufbereiteten Daten und wendet einen der drei Algorithmen an. Abbildung 18 zeigt die Verarbeitung der Daten mit Hilfe

des Algorithmus Local Outlier Factor in Kombination mit Datenset B. Der Quellcode und die Ergebnisse für die Algorithmen Isolation Forest und One Class SVM sowie die Resultate des Datenset A werden im Anhang bereitgestellt.

```
1  from sklearn.neighbors import LocalOutlierFactor
2
3  def do_lof(X):
4      clf = LocalOutlierFactor(n_neighbors=5)
5
6      pred = clf.fit_predict(X)
7      scores = clf.negative_outlier_factor_
8
9      return pred, scores
```

Abbildung 18: Anwenden des Algorithmus Local Outlier Factor <training.py> [1-9]

Zu Beginn wird das Model des Algorithmus definiert. Hierbei stellt der Parameter „n_neighbors“ die in der Berechnung betrachtete Größe der k-nächsten Nachbarn dar und kann je nach Anwendungsfall und Datensatz angepasst werden. Anschließend wird das Model mittels der Methode „fit_predict“ auf den gegebenen Daten trainiert und liefert die Klassifikation beziehungsweise Labels für die einzelnen Datenpunkte zurück. Das Ergebnis wird in der Variable „pred“ abgespeichert. Hierbei spiegelt das Label mit dem Wert 1 ein normales Datenobjekt und der Wert -1 einen Ausreißer wider. Darüber hinaus werden mit Hilfe der Funktion „negative_outlier_factor“ die Outlier Scores für die Daten in der Variablen „scores“ abgelegt. Dabei repräsentieren Werte nahe -1 normale Objekte, wohingegen Ausreißer einen deutlich höheren Score aufweisen. (scikit learn, 2020)

5. Ergebnisse

Nachdem der Algorithmus angewandt und die Ergebnisse in Variablen aufbewahrt wurden, können anschließend die bereits aufbereiteten Rohdaten mit den Analyseergebnissen zusammengeführt werden. Diesbezüglich werden im Dataframe zwei neue Spalten ‚scores' und ‚anomaly' eingeführt, welche mit den Ergebnissen des Algorithmus gefüllt werden. Die Spalte ‚anomaly' repräsentiert die Klassifikation während ‚scores' die Outlier Scores darstellt. In folgender Abbildung ist ein beispielhaftes Ergebnis des LOF Algorithmus mit N=10 Datenpunkten dargestellt.

	Anwender	Saldo	scores	anomaly
0	2.0	14093.81	-1.075862	1
1	0.0	24019.27	-0.980101	1
2	0.0	-26543.68	-0.982558	1
3	0.0	20714.57	-1.028000	1
4	0.0	26543.68	-1.046079	1
5	0.0	26543.68	-1.046079	1
6	0.0	-26543.68	-0.982558	1
7	6.0	14601.02	-1.320949	1
8	6.0	-30458.00	-1.857683	-1
9	6.0	30458.00	-1.078017	1

Abbildung 19: Ergebnisse des LOF Algorithmus in Tabellenform (N=10 Datenpunkte)

Zusätzlich wird ein Plot der Ergebnisse angezeigt. Dieser beinhaltet auf der X-Achse die Anwender und auf der Y-Achse den dazugehörigen Anomaly Score des Datenobjekts. Je niedriger der Score des Datenpunkts, desto stärker hat der Algorithmus diesen als Ausreißer eingeordnet. Für den Algorithmus LOF ist das Ergebnis in folgender Grafik dargestellt.

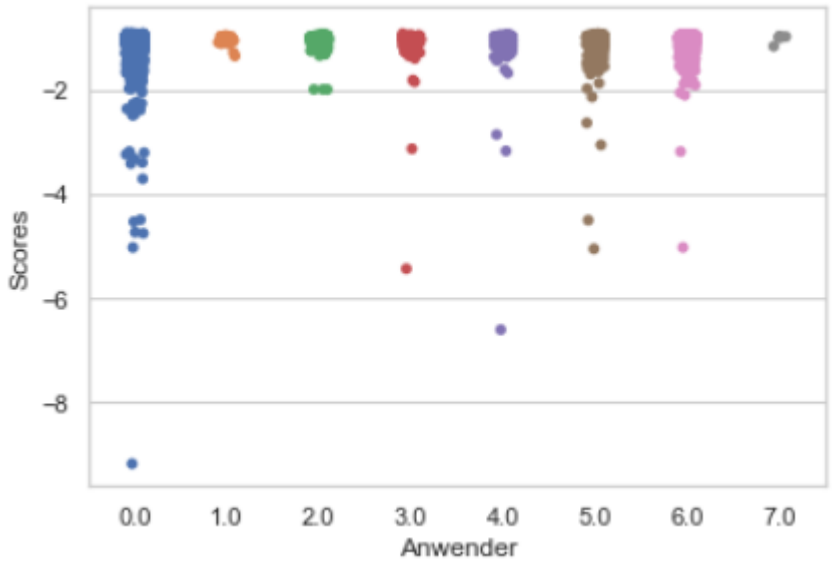


Abbildung 20: Ergebnisse des LOF Algorithmus als Plot

Darüber hinaus wurden mit Hilfe eines von Goix bereitgestellten Pakets im Modul Evaluation.py die ausgewählten Metriken zur Überprüfung der Leistung der Algorithmen, Excess Mass und Mass Volume berechnet und als Kurven dargestellt.

Hierbei dient die Spalte ‚scores' als Input für die Kalkulation der Metriken. Zunächst sind in folgenden zwei Tabellen die Werte der Metriken EM und MV der drei ausgewählten Algorithmen basierend auf Datenset A und B abgebildet.

	MV	EM
LOF	1,108e+06	4,104e-09
Isolation Forest	**9,680e+04**	**2,841e-08**
One Class SVM	1,099e+06	4,901e-09

Tabelle 4: Excess Mass und Mass Volume Metrik für Datenset A

	MV	EM
LOF	**1,588e+05**	1,955e-08
Isolation Forest	8,597e+04	2,281e-08
One Class SVM	1,845e+05	**2,285e-08**

Tabelle 5: Excess Mass und Mass Volume Metrik für Datenset B

Für Datenset A schneidet der Algorithmus Isolation Forest sowohl in der Metrik Mass Volume als auch bei der Metrik Excess Mass am besten ab. Darüber hinaus unterscheiden sich die Werte stark von denen der anderen Algorithmen. Daraus lässt sich ableiten, dass Isolation Forest für dieses Datenset im gegebenen Szenario die beste Leistung bringt.

Bei Datenset B kann kein Algorithmus eindeutig identifiziert werden, der in den gegebenen Metriken das beste Ergebnis erzielt. Für Mass Volume erreicht der Lokal Outlier Faktor den optimalen Wert. Außerdem liegt der Algorithmus One Class SVM bei der Berechnung der EM leicht vorn. Wobei sich hier die Werte zwischen Isolation Forest und One Class SVM lediglich marginal unterscheiden. In den folgenden beiden Abbildungen sind die MV- und EM Kurven für Datenset B dargestellt. Die Ergebnisse für Datenset A werden dem Anhang beigefügt.

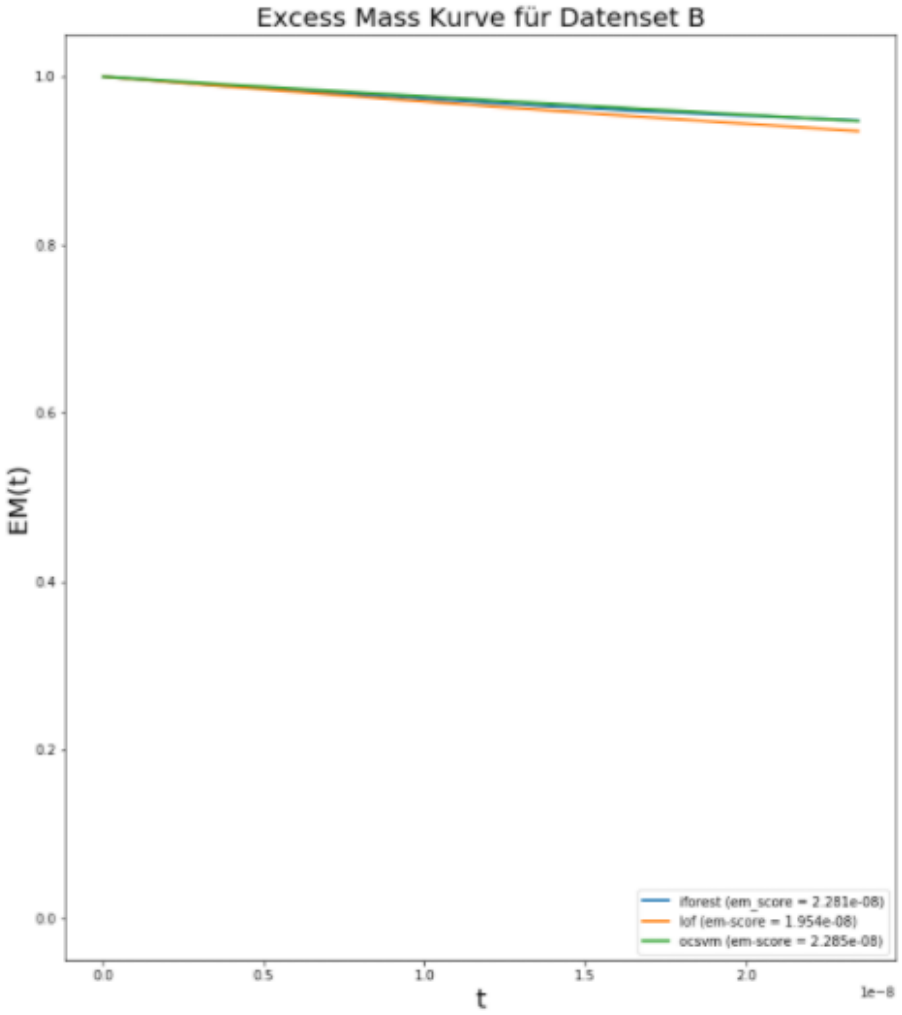


Abbildung 21: Excess Mass Kurve für Datenset B

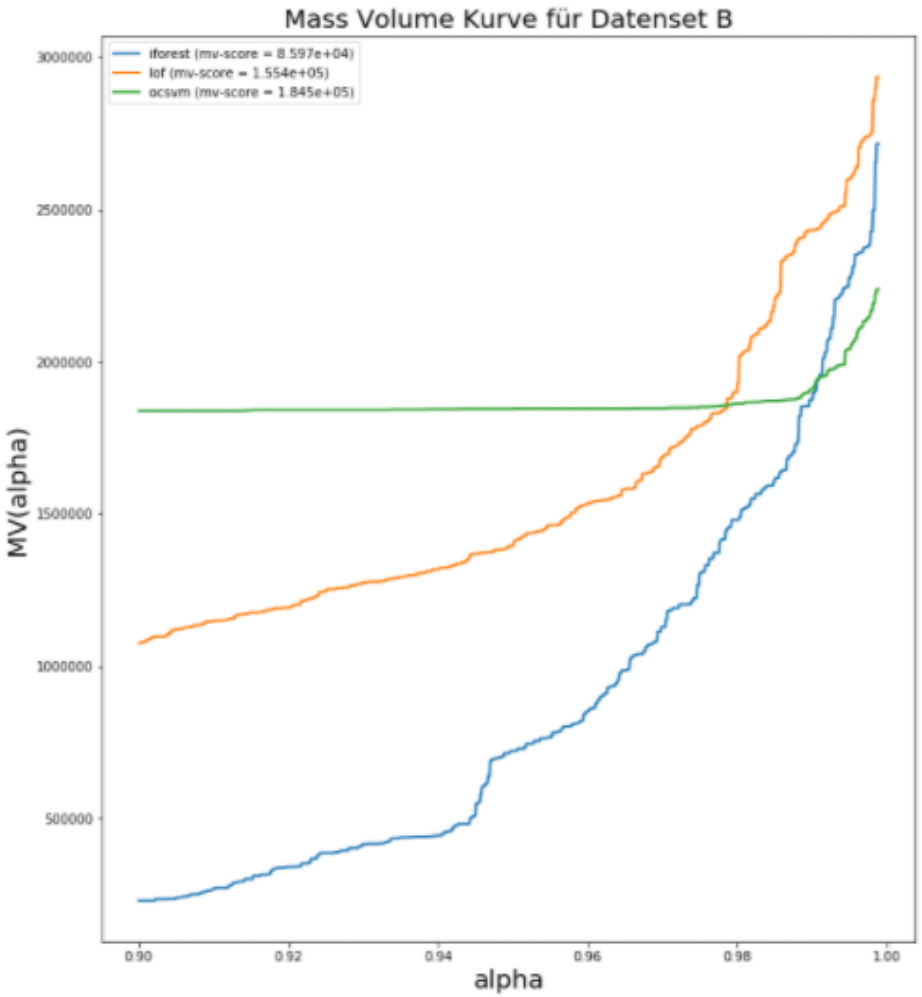


Abbildung 22: Mass Volume Kurve für Datenset B

Basierend auf den Ergebnissen kann kein Algorithmus zur Ausreißererkennung im betrachteten Anwendungsfall als am besten oder als am geeignetsten bewertet werden. Für jeden gegebenen Datensatz muss dies neu evaluiert und überprüft werden. Darüber hinaus werden die Metriken und somit auch die Auswahl des Algorithmus durch Veränderung der Parameter oder das Hinzufügen beziehungsweise das Weglassen von weiteren Features beeinflusst.

6. Zusammenfassung und Limitationen

Das Ziel dieser Arbeit ist es, festzustellen inwiefern die tägliche Arbeit des Wirtschaftsprüfers durch Machine Learning unterstützt werden kann. Die Ergebnisse der Untersuchung haben gezeigt, dass Auffälligkeiten und ungewöhnliche Werte in den Daten mit Hilfe unterschiedlicher Methoden der Ausreißererkennung identifiziert werden. Diese können dem Abschlussprüfer Hinweise auf mögliche Betrugsfälle oder ungewöhnliche Abläufe der Geschäftsprozesse geben. Anschließend müssen Wirtschaftsprüfer weitere Untersuchungen und Analysen durchführen, um Betrugsfälle letztendlich vollständig aufzudecken.

Es muss jedoch berücksichtigt werden, dass sich diese Forschung ausschließlich auf einen bestimmten Themenbereich konzentriert. Der in der Arbeit durchgeführte Ansatz stellt lediglich einen kleinen Teil des Gesamtprozesses dar.

Außerdem müssen zum einen weitere Kontenbereiche hinzugefügt, zum anderen zusätzliche Features in Betracht gezogen werden. Ferner sollten einerseits zusätzliche Unternehmen und andererseits weitere Anwendungsfälle in der Untersuchung validiert und berücksichtigt werden.

Zwar wird dem Abschlussprüfer durch die Anwendung der vorgestellten Methoden geholfen, jedoch kann hiermit lediglich auf Unregelmäßigkeiten und Auffälligkeiten aufmerksam gemacht werden. Es bedarf weiterhin einer manuellen und intensiven Untersuchung durch einen Experten.

Die Beschaffung von gelabelten Daten, die darüber hinaus Betrugsfälle beinhalten, stellt sich im Anwendungsgebiet der Wirtschaftsprüfung als außerordentlich schwierig heraus. Daher leistet die Arbeit einen Beitrag zu den von Goix (2016) vorgestellten Evaluierungsmöglichkeiten von ungelabelten Daten. Die Ergebnisse der Untersuchung haben basierend auf einem realen Anwendungsfall bestätigt, dass Excess Mass und Mass Volume zur Überprüfung der Leistung von unüberwachten Machine Learning Modellen dienen kann. Des Weiteren gewinnen unüberwachte Modelle aufgrund des geringen Anteils an gelabelten Daten in der Forschung sowie in der Praxis immer mehr an Bedeutung.

Jedoch werfen die gewonnenen Ergebnisse Fragen auf, die durch weitere Untersuchungen, beispielsweise die Betrachtung von neuen Anwendungsfällen und Algorithmen insbesondere neuronale Netze, ergänzt werden können. Ferner sollte der Fokus auf der Einbeziehung weiterer Datenquellen der Wirtschaftsprüfer und Corporate Governance gelegt werden, um Betrug aufzudecken. Zusätzlich können die Forschungsergebnisse dieser Arbeit in weiteren Bereichen abseits der Betrugserkennung hilfreich sein. So kann der vorgestellte Ansatz in das Preprocessing

für weitere Machine Learning Modelle integriert werden. Ausreißer können die Ergebnisse der Modelle und Untersuchungen stark beeinflussen. Die Ausreißererkennung kann als Datenbereinigung dazu dienen, Outlier zu identifizieren. Diese können anschließend aus dem Datenset entfernt werden, um eine gute Performance der Machine Learning Modelle zu gewährleisten.

Abschließend lässt sich zusammenfassen, dass der in der Forschung vorgestellte Ansatz als erster Schritt zur Aufdeckung von Betrugsfällen in der Wirtschaftsprüfung verwendet werden kann. Die Vorgehensweise wird empfohlen, falls große Datenmengen nicht vorhanden sind. Der erörterte Ansatz kann daher vor allem bei kleinen und mittelständischen Unternehmen angewandt werden und sollte Schritt für Schritt in die Prozesse und Analysen der Software für Wirtschaftsprüfer integriert werden.

Literaturverzeichnis

Achtert, E., Kriegel, H.-P., Reichert, L., Schubert, E., Wojdanowski, R., & Zimek, A. (2010). Visual Evluation of Outlier Detection Models. In H. Kitagawa, Y. Ishikawa, Q. Li, & C. Watanabe, *Database Systems for Advanced Applications* (S. 396-399). doi:10.1007/978-3-642-12098-5_34

Agyemang, M., & Ezeife, C. I. (2004). *Lsc-mine: Algorithm for mining local outliers.* Abgerufen am 6. August 2020 von https://www.semanticscholar.org/paper/LSC-Mine-%3A-Algorithm-for-Mining-Local-Outliers-Agyemang-Ezeife/9135f051752b3e6b694d640ee1a060d33317c24a

AIMultiple. (12. Juli 2020). Machine Learning Accuracy: Learn the Metric to Assess ML Models. Abgerufen am 16. August 2020 von https://research.aimultiple.com/machine-learning-accuracy/

Alpaydin, E. (2009). *Introduction to Machine Learning.* The MIT Press. doi:10.5555/1734076

Anton, S. D., Kanoor, S., Fraunholz, D., & Schotten, H. D. (2018). Evaluation of Machine Learning-based Algorithms on an Industrial Modbus/TCP Data Set. *Proceedings of the 13th International Conference on Availability, Reliability and Security*. doi:10.1145/3230833.3232818

Barnett, V., & Lewis, T. (März 1995). Outliers in statistical data. *International Journal of Forecasting*, S. 175-176. doi:10.2307/2066277

Bauer, J., & Gross, J. (Januar 2010). Difficulties Detecting Fraud? The Use of Benford's Law on Regression Tales. *Jahrbücher für Nationalökonomie und Statistik*, S. 733-748. doi:10.1515/jbnst-2011-5-611

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (Juli 1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedngs of the Fifth Annual Workshop on Computational Learning Theory*, S. 144-152. doi:10.1145/130385.130401

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (16-18. May 2000). LOF: Identifying Density-Based Local Outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, S. 93-104. doi:10.1145/342009.335388

Bussmann, P. D.-D., Nestler, C., & Salvenmoser, S. (Februar 2018). *pwc.de*. Abgerufen am 31. Juli 2020 von pwc.de: https://www.pwc.de/de/risk/pwc-wikri-2018.pdf

Chan, C. (5. Juli 2018). What is a ROC Curve and How to Interpret It. Abgerufen am 27. August 2020 von https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/

Chandola, V., Banerjee, A., & Kumar, V. (September 2009). Anomaly Detection: A Survey. *ACM Computing Surveys*. doi:10.1145/1541880.1541882

Chepenko, D. (15. September 2018). A Density-based algorithm for outlier detection. Abgerufen am 14. September 2020 von https://towardsdatascience.com/density-based-algorithm-for-outlier-detection-8f278d2f7983

Clémençon, S., & Jakubowicz, J. (April 2013). Scoring anomalies: a M-estimation formulation. *Proceedings of Machine Learning Research*. Abgerufen am 14. August 2020 von http://proceedings.mlr.press/v31/clemencon13a.pdf

Clémencon, S., & Thomas, A. (Mai 2017). Mass Volume Curves and Anomaly Ranking. *Electronic Journal of Statistics*. doi:10.1214/18-EJS1474

cloudfactory. (2020). The Ultimate Guide to Data Labeling for Machine Learning. Abgerufen am 21. September 2020 von https://www.cloudfactory.com/data-labeling-guide

DATEV eG. (2020). DATEV Datenprüfung hilft bei Journal Entry Testing. Abgerufen am 18. August 2020 von https://www.datev.de/web/de/aktuelles/datev-news/datenpruefung-comfortclassic-hilft-bei-journal-entry-testing-jet/

Deepika Pahuja, R. Y. (März 2013). Outlier Detection for Different Applications: Review. *International Journal Of Engineering Research & Technology*, S. 12. Abgerufen am 1. September 2020

Diekmann, A. (April 2007). Not the First Digit! Using Benford's Law to Detect Fraudulent Scientific Data. *Journal of Applied Statistics*, S. 321-329. doi:10.1080/02664760601004940

Diekmann, A., & Jann, B. (August 2010). Benford's Law and Fraud Detection: Facts and Legends. *German Economic Review*, S. 397-401. doi:10.1111/j.1468-0475.2010.00510.x

Dixon, W. J. (Dezember 1950). Analysis of Extreme Values. *The Annals of Mathematical Statistics*, S. 488-506. doi:10.1214/aoms/1177729747

Domingues, R., & Filippone, M. (September 2017). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*. doi:10.1016/j.patcog.2017.09.037

Durtschi, C., Hillison, W., & Pacini, C. (Januar 2004). The Effective Use of Benford's Law to Assist in Detection Fraud in Accounting Data. *Journal of Forensic Accounting*. Abgerufen am 6. August 2020 von https://www.researchgate.net/publication/241401706_The_Effective_Use_of_Benford's_Law_to_Assist_in_Detecting_Fraud_in_Accounting_Data

EBS audit & accounting. (2020). Journal Entry Testing as a Service (JETaaS). Abgerufen am 18. August 2020 von https://www.ebs-audit.com/journal-entry-testing-as-a-service-jetaas

Emmott, A., Das, S., Dietterеich, T., Fern, A., & Wong, W.-K. (26. August 2016). *A Meta-Analysis of the Anomaly Detection Problem.* Abgerufen am 14. August 2020

Fawcett, T. (Januar 2003). ROC Gaphs: Notes and Practical Considerations for Data Mining Researchers. *HP Labs*. Abgerufen am 18. August 2020 von https://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf

Fawcett, T. (Juni 2006). An introduction to ROC analysis. *Pattern Recognition Letters*, S. 861-874. doi:10.1016/j.patrec.2005.10.010

Goix, N. (Juli 2016). How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms? Abgerufen am 14. August 2020 von https://www.researchgate.net/publication/304859477_How_to_Evaluate_the_Quality_of_Unsupervised_Anomaly_Detection_Algorithms

Goix, N., Sabourin, A., & Clémencon, S. (Februar 2015). On Anomaly Ranking and Excess-Mass Curves. Abgerufen am 14. August 2020 von https://www.researchgate.net/publication/271854939_On_Anomaly_Ranking_and_Excess-Mass_Curves

Goldstein, M., & Uchida, S. (April 2016). A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLoS ONE*, S. 1-31. doi:10.1371/journal.pone.0152173

Google LLC. (2020). Classification: ROC Curve and AUC. Abgerufen am 27. August 2020 von https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Google LLC. (2020). Framing: Key ML Terminology. Abgerufen am 14. September 2020 von https://developers.google.com/machine-learning/crash-course/framing/ml-terminology

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques.* doi:10.1016/C2009-0-61819-5

Hand, D. J., & Adams, N. M. (22. Juni 2015). Data Mining. *WilyStatsRef: Statistics Reference Online*, S. 7. doi:10.1002/9781118445112.stat06466.pub2

Hanley, J. A., & McNeil, B. J. (April 1982). The Meaning and Use of the Area under a Receiver Operating Characterisitc (ROC) Curve. *Radiology Society of North America*, S. 29-35. doi:10.1148/radiology.143.1.7063747

Hawkins, D. (1980). *Identification of Outliers*. Springer Netherlands. doi:10.1007/978-94-015-3994-4

Hevner, A. R., March, S. T., Ram, S., & Park, J. (März 2004). Design Science in Information Systems Research. *MIS Quarterly*, S. 75-105. Abgerufen am 15. September 2020 von https://www.researchgate.net/publication/201168946_Design_Science_in_Information_Systems_Research

Hewahi, N., & Saad, M. (März 2007). Class Outliers Mining: Distance-Based Approach. *International Journal of Intelligent Systems Technologies and Applications*. Abgerufen am 6. August 2020

Hodge, V., & Austin, J. (October 2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, S. 85-126. Abgerufen am 2. September 2020 von https://www.researchgate.net/publication/220638052_A_Survey_of_Outlier_Detection_Methodologies

Hofmann, D. S. (2008). *Handbuch Anti-Fraud-Management Bilanzbetrug erkennen - vorbeugen - bekämpfen*. Erich Schmidt Verlag. Abgerufen am 31. Juli 2020 von https://bilder.buecher.de/zusatz/23/23506/23506828_inha_1.pdf

Huang, F. J., & LeCun, Y. (17-22. Juni 2006). Large-scale learning with SVM and convolutional nets for generic object categorization. *Computer Society Conference on Computer Vision and Pattern Recognition*. doi:10.1109/CVPR.2006.164

Institut der Wirtschaftsprüfer. (2006). *IDW Prüfungsstandard: Zur Aufdeckung von Unregelmäßigkeiten im Rahmen der Abschlussprüfung (IDW PS210)*. Institut der Wirtschaftsprüfer.

Jain, A. K. (Juni 2010). Data Clustering: 50 years beyond K-means. *Pattern Recognition Letters*, S. 651-666. doi:10.1016/j.patrec.2009.09.011

Kanungo, T., Netanyahu, N. S., & Wu, A. Y. (Juli 2002). An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *Transcations on Pattern*

Analysis and Machine Intelligence, S. 881-891. doi:10.1109/TPAMI.2002.1017616

Knorr, E. M., & Ng, R. T. (27. August 1998). Algorithms for Mining Distance-Based Outliers in Large Datasets. *Proceedings of 24rd International Conference on Very Large Data Bases*, S. 392-304. Abgerufen am 19. August 2020

Kriegel, H.-P., Kröger, P., & Zimek, A. (2010). Outlier Detection Techniques. München, Bayern, Deutschland. Abgerufen am 19. August 2020 von https://archive.siam.org/meetings/sdm10/tutorial3.pdf

Kriegel, H.-P., Kröger, P., Schubert, E., & Zimek, A. (November 2009). LoOP: Local Outlier Probabilities. *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM*, S. 2-6. doi:10.1145/1645953.1646195

Kriegel, H.-P., Kröger, P., Schubert, E., & Zimek, A. (April 2009). Outlier Detection in Axis-Parallel Subsapces of High Dimensional Data. *13th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, S. 27-30. doi:10.1007/978-3-642-01307-2_86

Lasko, T. A., Bhagwat, J. G., Zou, K. H., & Ohno-Machado, L. (Oktober 2005). The use of receiver operating characteristic curves in biomedical informatics. *Journal of Biomedical Informatics*, S. 404-413. doi:10.1016/j.jbi.2005.02.008

Li, K.-K., Huang, H.-K., Tian, S.-F., & Xu, W. (2-5. November 2003). Improving one-class SVM for anomaly detection. *Machine Learning and Cybernetics*, S. 3077-3081. doi:10.1109/ICMLC.2003.1260106

Liu, S., Wang, X., Liu, M., & Zhu, J. (März 2017). Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, S. 48-56. doi:10.1016/j.visinf.2017.01.006

Lui, F. T., Ting, K. M., & Zhou, Z. (15-19. Dezember 2008). Isolation Forest. *International Conference on Data Mining*. doi:10.1109/icdm.2008.17

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of Machine Learning*. The MIT Press. Abgerufen am 16. August 2020

Müller, A. C., & Guido, S. (2017). *Einführung in Machine Learning mit Python*. O'Reilly Verlag. Abgerufen am 18. August 2020

Narkhede, S. (26. Juni 2018). Understanding AUC - ROC Curve. Abgerufen am 19. August 2020 von https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

pandas. (2020). pandas-docs. Abgerufen am 21. September 2020 von https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

Petrovskiy, M. I. (2003). Outlier Detection Algorithms in Data Mining Systems. *Programming and Computer Software*, S. 228-237. Abgerufen am 8. August 2020 von https://cs.nju.edu.cn/zhouzh/zhouzh.files/course/dm/reading/reading06/petrovskiy_pcs03.pdf

Pfeffers, K., Gengler, C., Tuunanen, T., & Rossi, M. (Februar 2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, S. 45-77. doi:10.2307/40398896

Ramaswamy, S., Rastogi, R., & Shim, K. (16. Mai 2000). Effiecient algorithms for mining outliers from large data sets. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, S. 427-438. doi:10.1145/335191.335437

Reitmaier, T. (2015). *Aktives Lernen für Klassifikationsprobleme unter der Nutzung von Strukturinformationen.* Kassel: kassel university press GmbH. Abgerufen am 26. August 2020

Rousseeuw, P. J., & Leroy, A. M. (1987). Robust Regression and Outlier Detection. In *Wiley Series in Probability and Statistics.* Wiley. doi:10.1002/0471725382

Schmitt, J. (04. April 2017). *finance-magazine*. Abgerufen am 31. Juli 2020 von finance-magazine: https://www.finance-magazin.de/banking-berater/wirtschaftspruefer/welche-verantwortung-traegt-ey-beim-abb-betrug-1400601/

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., & Smola, A. J. (27. November 1999). Estimating the Support of a High-Dimensional Distribution. *Neural Computation*. doi:10.1162/089976601750264965

scikit learn. (2020). Abgerufen am 24. September 2020 von https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html

Singh, K., Malik, D., & Sharma, N. (April 2011). Evolving limitations in K-means algorithm in data mining and their removal. *International Journal of Computational Engieering & Management*, S. 105-109. Abgerufen am 22. August 2020

Tammaru, M., & Alver, L. (Januar 2016). Application of Benford's Law for Fraud Detection in Financial Statements: Theoretical Review. *5th International Conference on Accounting, Auditing and Taxation (ICAAT 2016)*. doi:10.2991/icaat-16.2016.46

Webster, J., & Watson, R. T. (2. Juni 2002). Analyzing the past to prepare for the future: writing a literatur review. *MIS Quarterly*, S. 23. Abgerufen am 1. August 2020

Zengyou, H., Xiaofei, X., & Shengchun, D. (Juni 2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, S. 1641-1650. doi:10.1016/S0167-8655(03)00003-5

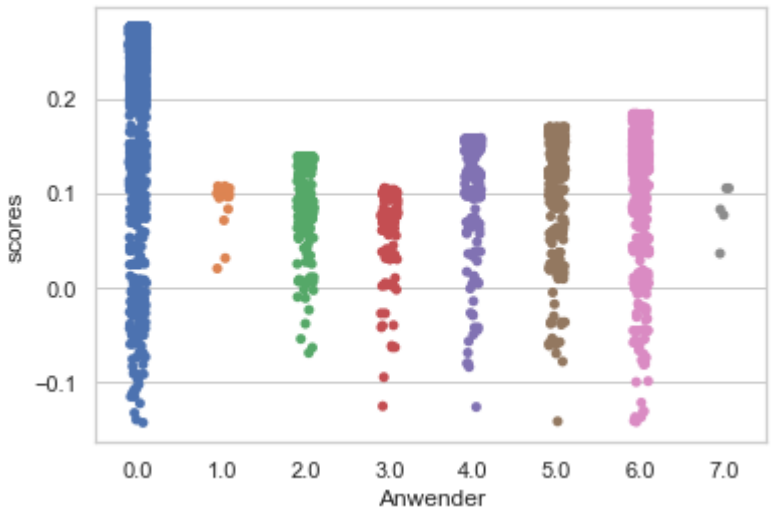
Anhang A Quellcode Isolation Forest und One Class SVM

```python
from sklearn.ensemble import IsolationForest

def do_IForest(X):
    clf = IsolationForest(n_estimators=50, max_samples='auto',
          contamination=float(0.01),max_features=2).fit(X)

    pred = clf.predict(X)
    scores = clf.decision_function(X)

    return pred,scores
```

```python
from sklearn.svm import OneClassSVM

def do_one_class_svm(X):
    clf = OneClassSVM(gamma=0.00001, nu=0.01).fit(X)

    pred = clf.predict(X)
    scores = clf.score_samples(X)

    return pred,scores
```

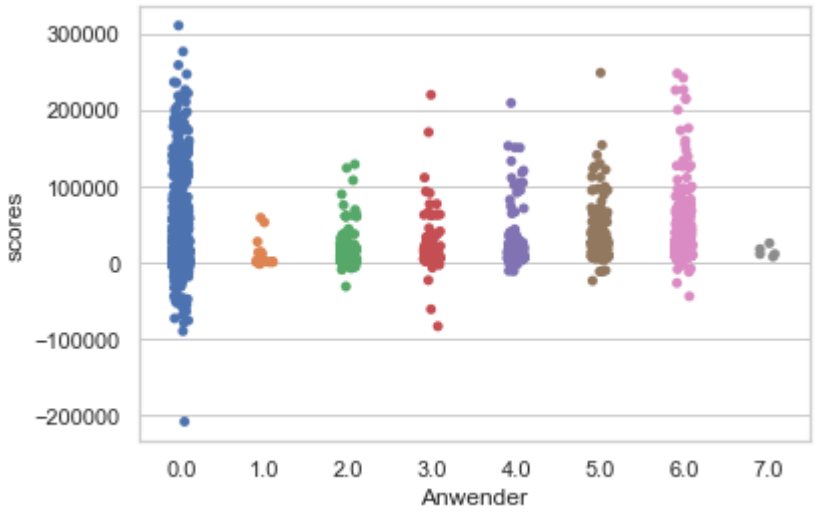
Anhang B Ergebnisse Isolation Forest für Datenset B

	Anwender	Umsaetze	scores	anomaly
0	2.0	14093.81	0.062163	1
1	0.0	24019.27	0.200549	1
2	0.0	-26543.68	0.005453	1
3	0.0	20714.57	0.193840	1
4	0.0	26543.68	0.189855	1
5	0.0	26543.68	0.189855	1
6	0.0	-26543.68	0.005453	1
7	6.0	14601.02	0.087434	1
8	6.0	-30458.00	-0.073747	-1
9	6.0	30458.00	0.015985	1

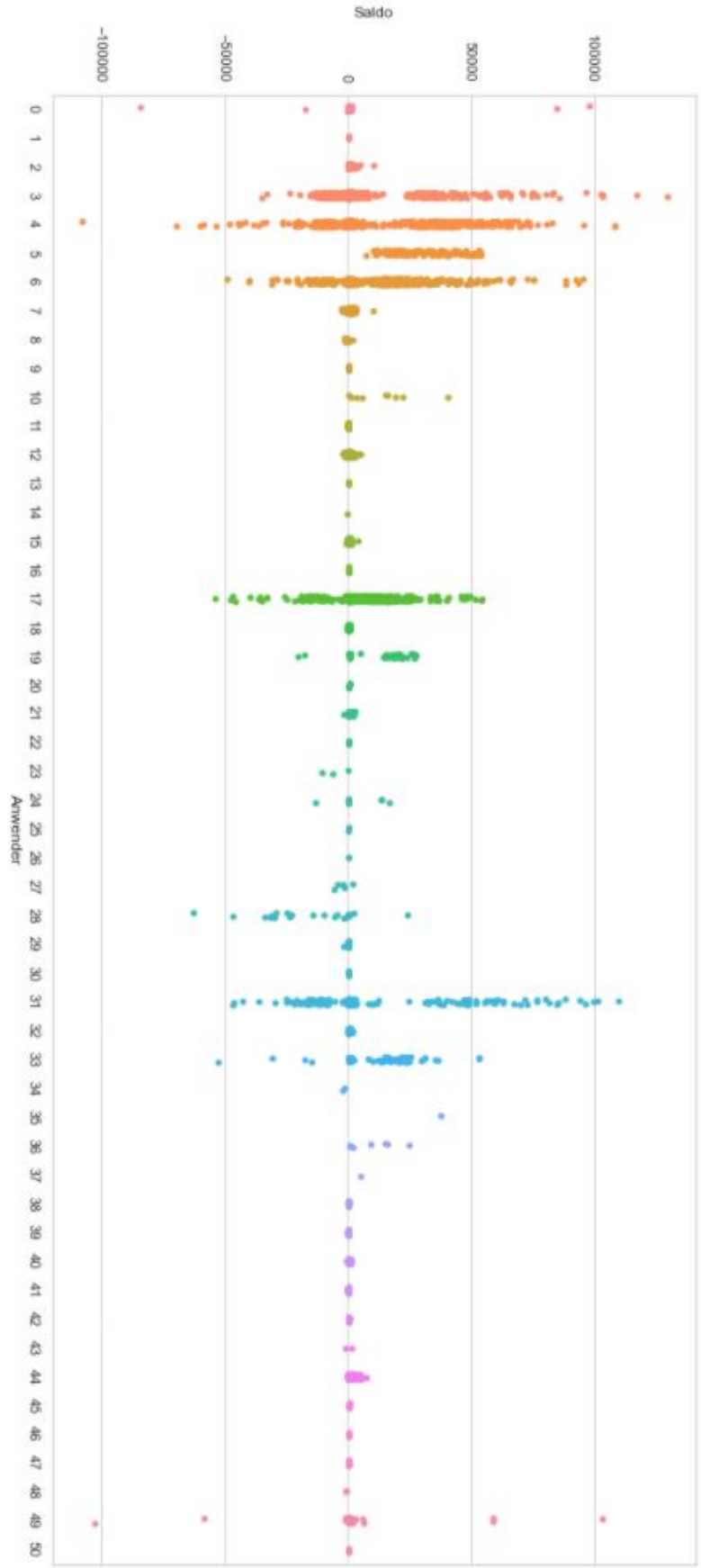


Anhang C Ergebnisse One Class SVM für Datenset B

	Anwender	Umsaetze	scores	anomaly
0	2.0	14093.81	20855.576568	1
1	0.0	24019.27	29294.159125	1
2	0.0	-26543.68	-32372.956588	-1
3	0.0	20714.57	25263.711581	1
4	0.0	26543.68	32372.956588	1
5	0.0	26543.68	32372.956588	1
6	0.0	-26543.68	-32372.956588	-1
7	6.0	14601.02	28807.404981	1
8	6.0	-30458.00	-26147.058699	-1
9	6.0	30458.00	48146.747706	1

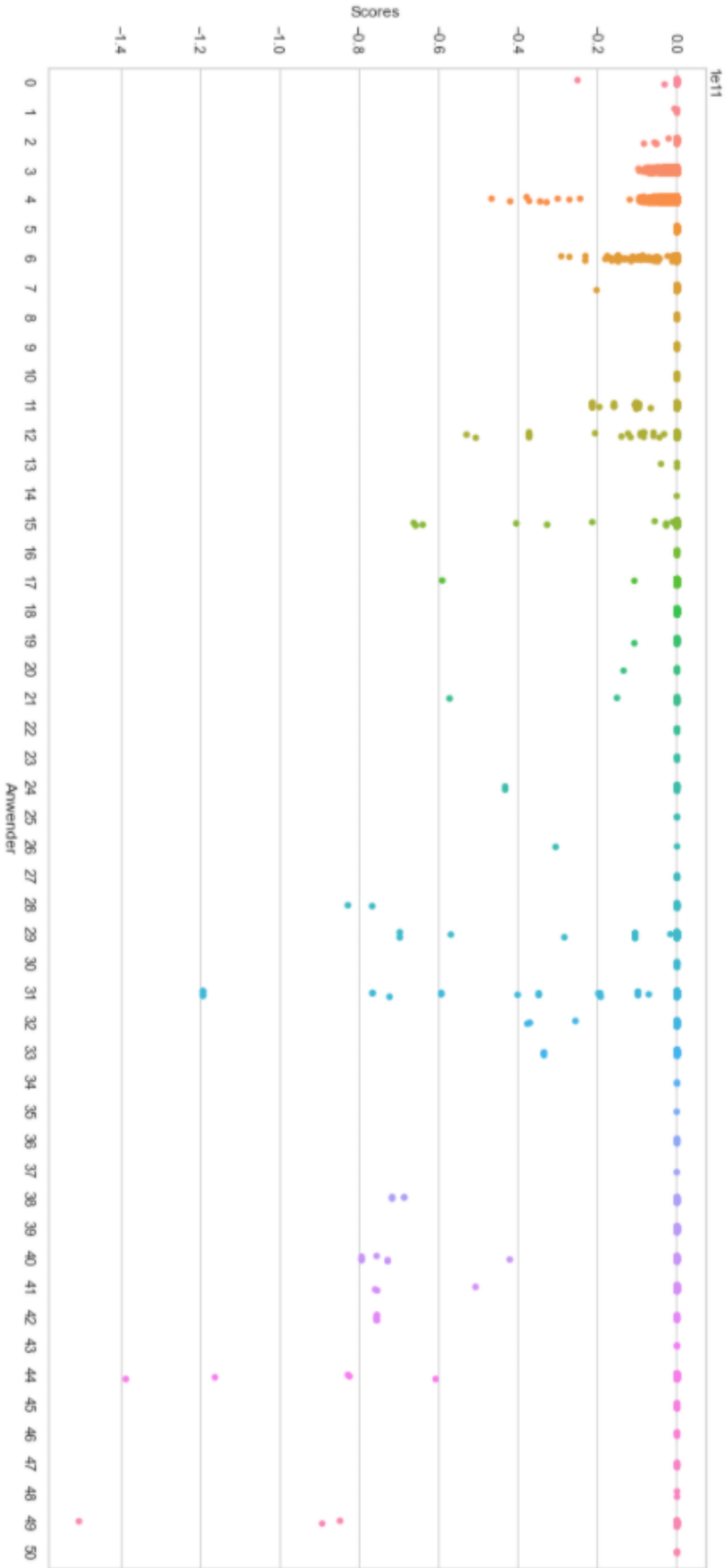


Anhang D Datenset A nach vollständigem Preprocessing



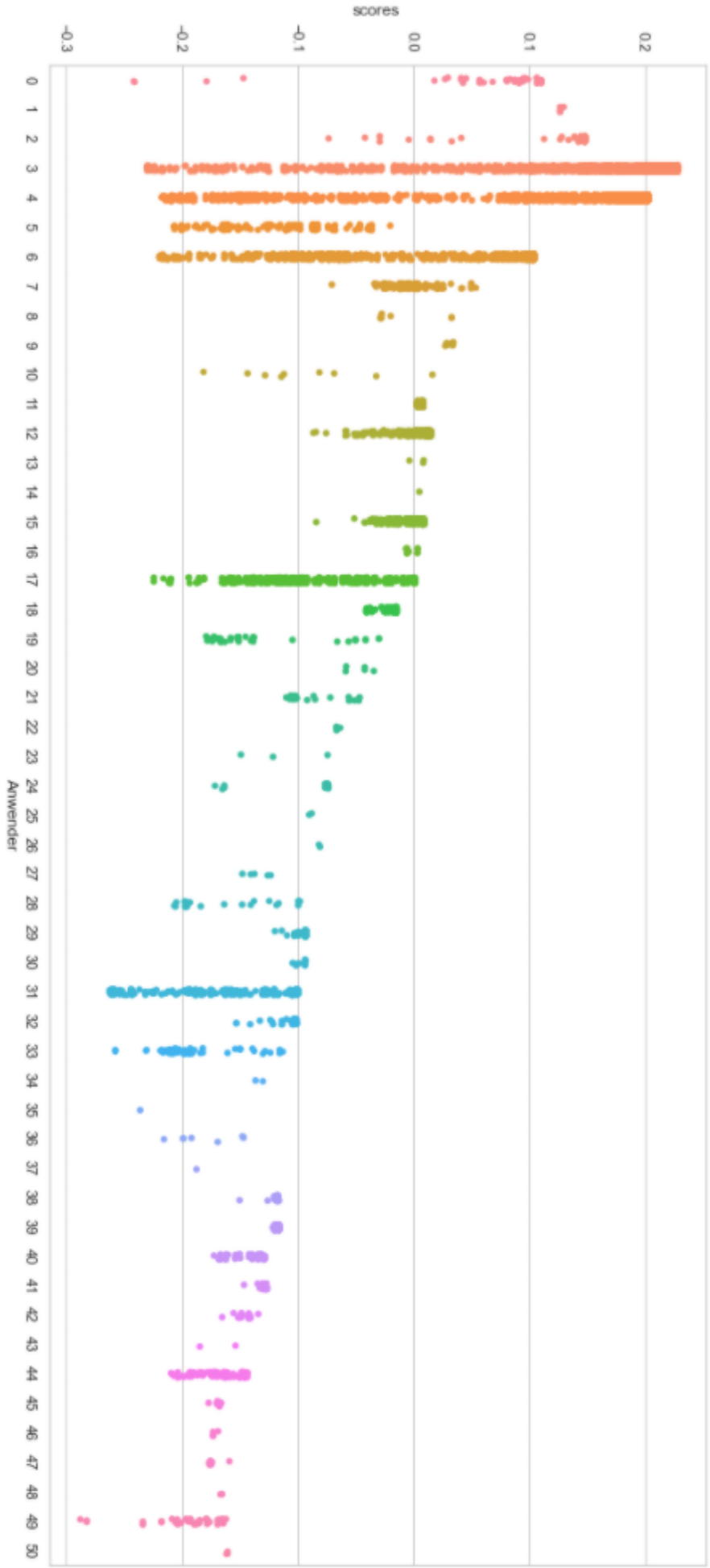
Anhang E Ergebnisse Local Outlier Factor für Datenset A

	Anwender	Saldo	Scores	anomaly
0	3	46.86	-1.008079e+00	1
1	3	293.66	-9.780515e-01	1
2	31	65.09	-1.849520e+00	-1
3	3	27.30	-1.325569e+00	1
4	4	4.05	-1.022851e+00	1
5	3	90.89	-1.288406e+00	1
6	3	38.34	-9.851002e-01	1
7	3	150.00	-1.002123e+00	1
8	3	1459.17	-1.040593e+00	1
9	4	6.09	-1.920600e+08	-1


Scores
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
-1.4
1e11
Anwender
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

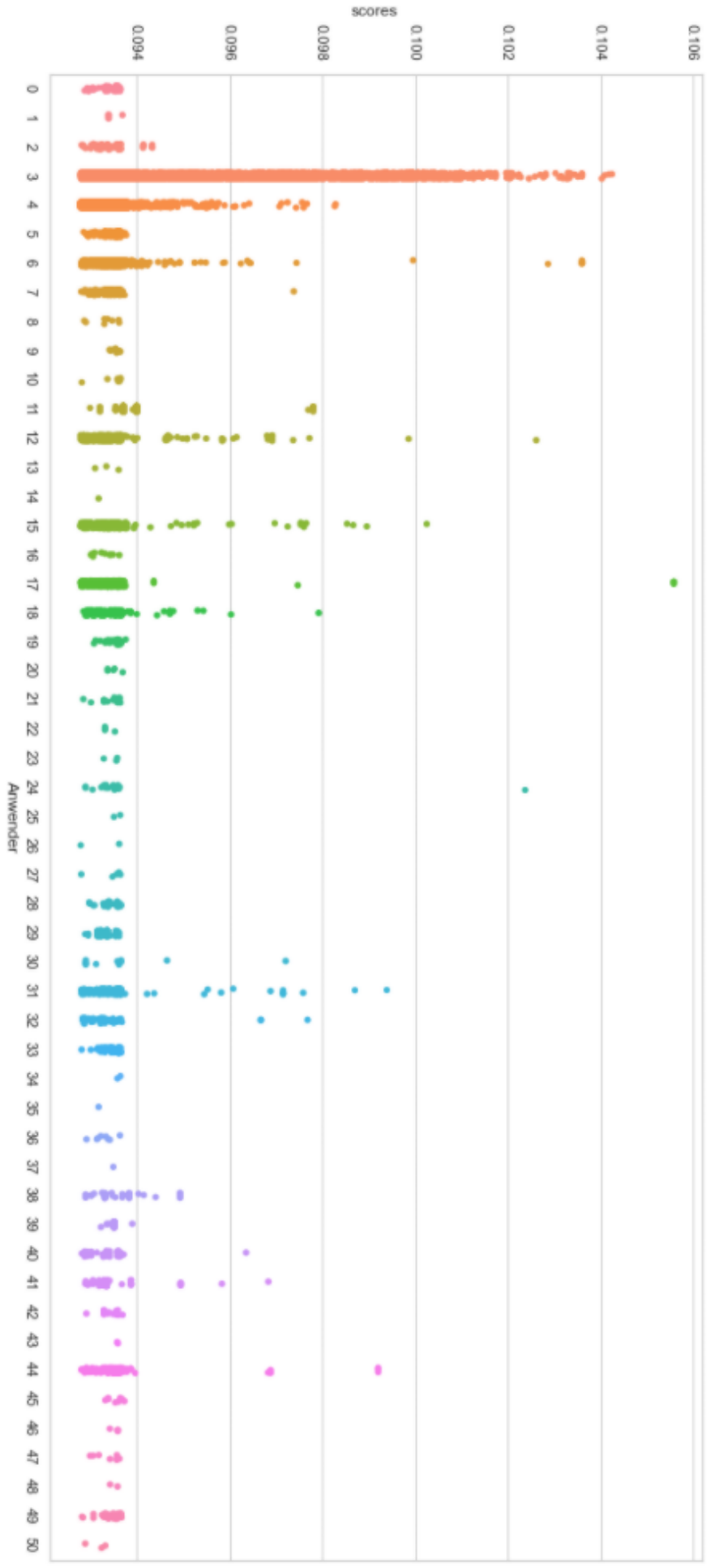
Anhang F Ergebnisse Isolation Forest für Datenset A

	Anwender	Umsaetze	scores	anomaly
0	3	46.86	0.226758	1
1	3	293.66	0.209323	1
2	31	65.09	-0.107739	-1
3	3	27.30	0.226829	1
4	4	4.05	0.201012	1
5	3	90.89	0.226069	1
6	3	38.34	0.226758	1
7	3	150.00	0.221980	1
8	3	1459.17	0.107738	1
9	4	6.09	0.201012	1

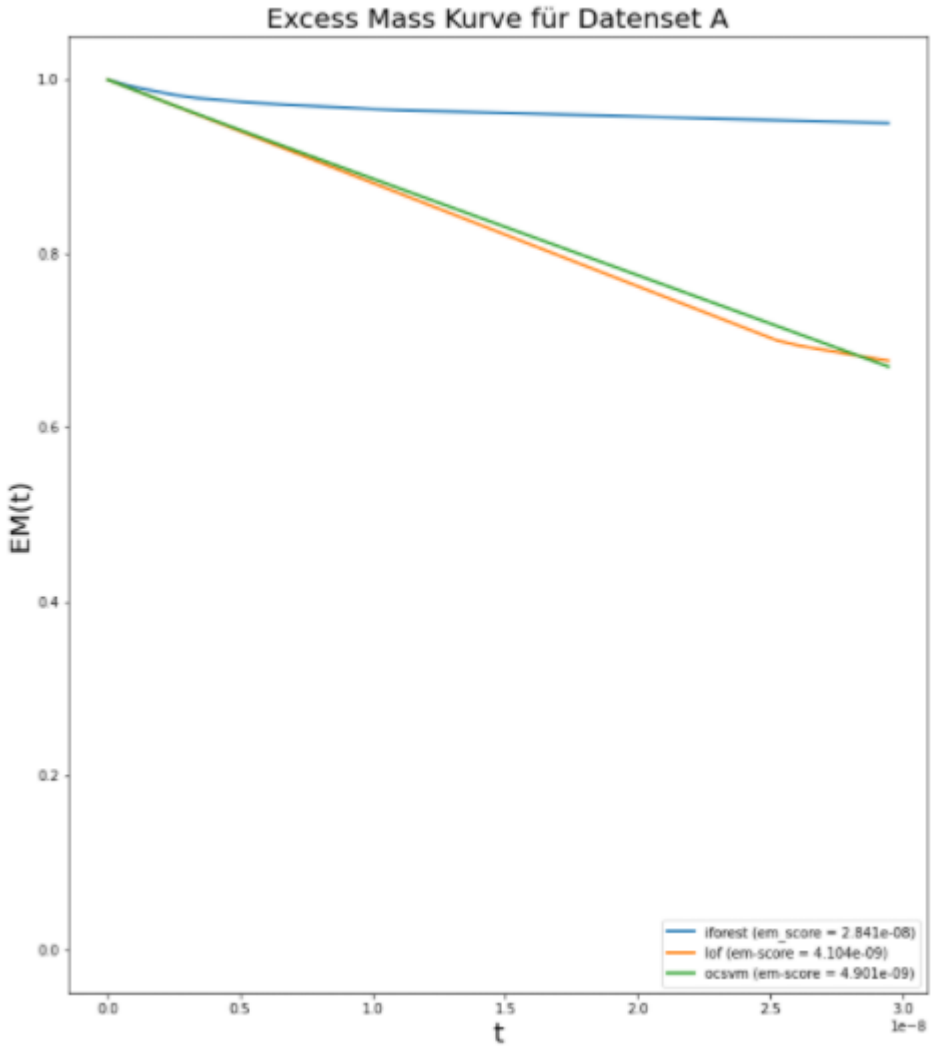

scores
-0.3
-0.2
-0.1
0.0
0.1
0.2
Anwender
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

Anhang G Ergebnisse One Class SVM für Datenset A

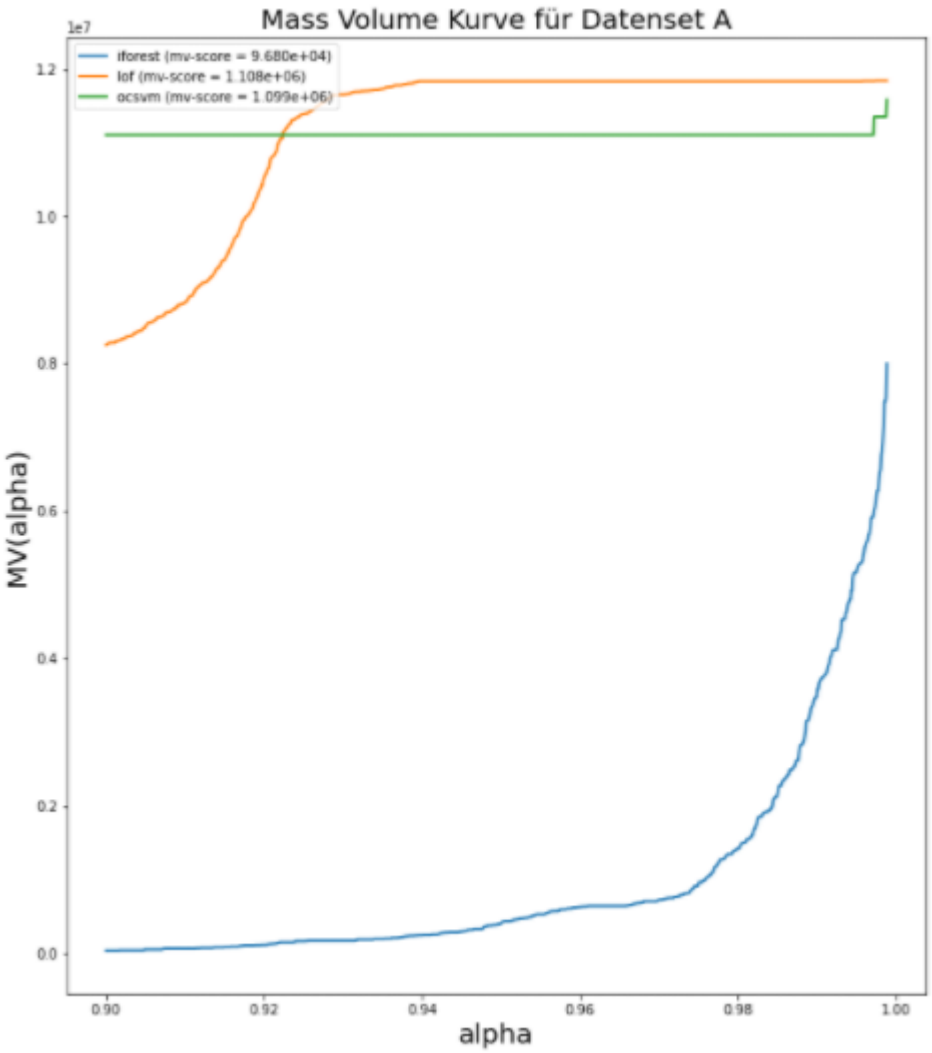
	Anwender	Umsaetze	scores	anomaly
0	3	46.86	0.093195	-1
1	3	293.66	0.093667	1
2	31	65.09	0.093576	1
3	3	27.30	0.095176	1
4	4	4.05	0.093155	-1
5	3	90.89	0.095795	1
6	3	38.34	0.095387	1
7	3	150.00	0.093516	1
8	3	1459.17	0.093282	-1
9	4	6.09	0.092843	-1


scores
0.094
0.096
0.098
0.100
0.102
0.104
0.106
Anwender
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

Anhang H Excess Mass und Mass Volume Kurve für Datenset A



Excess Mass und Mass Volume Kurve für Datenset A



Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit „Unregelmäßigkeiten in der Abschlussprüfung – Fraud Detection mit Hilfe von Machine Learning“ selbständig und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch an keiner anderen Prüfungsbehörde vorgelegen hat.

Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

_____ _____

Ort, Datum Unterschrift